

**SanUSB**

Robótica e Automação



# CONSTRUÇÃO DE ROBÔS MÓVEIS EDUCACIONAIS DE BAIXO CUSTO



[www.tinyurl.com/SanUSB](http://www.tinyurl.com/SanUSB)

[www.youtube.com/user/Imaculada27](http://www.youtube.com/user/Imaculada27)

[sanusb\\_laese@yahoo.com.br](mailto:sanusb_laese@yahoo.com.br)

[www.youtube.com/user/sandrojuca](http://www.youtube.com/user/sandrojuca)



## ÍNDICE

<b>1. INTRODUÇÃO À ROBÓTICA.....</b>	<b>PÁG.2</b>
<b>1.1 AS TRÊS LEIS FUNDAMENTAIS DA ROBÓTICA (SEGUNDO ISAAC ASIMOV).....</b>	<b>PÁG.2</b>
<b>1.2 CLASSIFICAÇÃO DE ROBÔS QUANTO À GERAÇÃO.....</b>	<b>PÁG.3</b>
<b>1.3 O QUE É UM ROBÔ?.....</b>	<b>PÁG.3</b>
<b>1.4 CLASSIFICAÇÃO DOS ROBÔS MÓVEIS.....</b>	<b>PÁG.4</b>
<b>1.5 DESAFIOS DA ROBÓTICA MÓVEL.....</b>	<b>PÁG.7</b>
<b>2. FERRAMENTA SanUSB.....</b>	<b>PÁG.8</b>
<b>2.1 UTILIZANDO A FERRAMENTA SanUSB.....</b>	<b>PÁG.9</b>
<b>2.2 GRAVANDO O MICROCONTROLADOR VIA USB NO WINDOWS.....</b>	<b>PÁG.14</b>
<b>2.3 GRAVANDO O MICROCONTROLADOR VIA USB NO LINUX.....</b>	<b>PÁG.16</b>
<b>3. ATUADORES APLICADOS EM ROBÓTICA.....</b>	<b>PÁG.17</b>
<b>3.1 ACIONAMENTO DE MOTORES CC (CORRENTE CONTÍNUA) DE BAIXA TENSÃO...PÁG.18</b>	
<b>3.2 ACIONAMENTO DE MOTORES DE PASSO.....</b>	<b>PÁG.34</b>
<b>3.3 SERVO-MOTORES.....</b>	<b>PÁG.37</b>
<b>4. SENSORES.....</b>	<b>PÁG.39</b>
<b>4.1 SENSORES DE PROXIMIDADE.....</b>	<b>PÁG.41</b>
<b>4.2 TRANSMISSOR E RECEPTOR INFRAVERMELHO (IR).....</b>	<b>PÁG.45</b>
<b>5. MONTAGEM DO ROBÔ MÓVEL DE BAIXO CUSTO.....</b>	<b>PÁG.47</b>

## 1. INTRODUÇÃO À ROBÓTICA

- **Mas meu senhor, eu sou robô.**
- Não preferia ser homem?
- **Gostaria era de ser o melhor cirurgião. O que não seria possível se eu fosse homem. Mas se eu pudesse apenas ser um robô mais aperfeiçoado, preferiria ser um robô mais aperfeiçoado.**
- Não se ofende com o fato de que eu posso lhe dar ordens? Obrigá-lo a levantar-se, sentar-se, andar para lá e para cá, apenas pedindo para que obedeça?
- **Tenho maior prazer em agradar o senhor. Se as suas ordens interferissem no meu comportamento em relação ao senhor ou a qualquer outro ser humano, eu não lhe obedeceria. Quanto ao mais, tenho o maior prazer em ser obediente. Agora, em quem eu devo efetuar a operação?**
- Em mim mesmo. Respondeu Andrew.
- **Mas isso é impossível. Trata-se, evidentemente, de uma operação prejudicial.**
- Não interessa. Afirmou Andrew calmamente.
- **Eu não posso causar danos. Retrucou o cirurgião.**
- Para uma criatura humana, claro que não pode. Disse Andrew, mas eu também sou robô.

---

Trecho do conto de ficção científica "O Homem Bicentenário" escrito por Isaac Asimov em 1976.

---

### 1.1 AS TRÊS LEIS FUNDAMENTAIS DA ROBÓTICA (SEGUNDO ISAAC ASIMOV):

- 1ª** - Um robô não pode causar dano a um ser humano nem, por omissão, permitir que um ser humano sofra;
- 2ª** - Um robô deve obedecer às ordens dadas por seres humanos, exceto quando essas ordens entrarem em conflito com a Primeira Lei;
- 3ª** - Um robô deve proteger sua própria existência, desde que essa proteção não se choque com a Primeira nem com a Segunda Lei da robótica.

---

Isaac Asimov nasceu numa pequena aldeia da União Soviética em 1920 e migrou com a família aos três anos de idade para os Estados Unidos onde viveu até sua morte, em 1992.

---

## 1.2 CLASSIFICAÇÃO DE ROBÔS QUANTO À GERAÇÃO

**1ª Geração** - são incapazes de obter qualquer informação sobre o meio. Podem realizar apenas movimentos pré-programados e as informações que eles retornam sobre o ambiente de operação é mínima.

**2ª Geração** - possui todas as características da 1ª Geração, acrescentando uma detalhada comunicação com seu ambiente. Esta comunicação é atingida através de sistemas de sensoriamento e identificação. Necessita de computadores mais velozes, com maior memória, e também um grande avanço na capacidade de sensoriamento.

## 1.3 O QUE É UM ROBÔ?

Um robô móvel é um dispositivo mecânico montado sobre uma base não fixa que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente.

Este sistema computacional pode ser um microcontrolador, semelhante ao que vamos estudar neste curso de Robótica Móvel Educacional; e que possui internamente os mesmos componentes que integram um computador: uma CPU (*Central Processor Unit*), memórias RAM (dados), flash (programa) e E2PROM, pinos de I/O (*Input/Output*), além de outros periféricos internos, tais como, osciladores, canal USB, interface serial assíncrona USART, módulos de temporização e conversores A/D, entre outros, integrados em um mesmo componente (chip).

A interação com o ambiente se dá através de ciclos de percepção-ação que consistem em três passos fundamentais: obtenção de informações acerca do ambiente através dos sensores; processamento das informações obtidas e seleção de ações que serão executadas; execução das ações planejadas por uma pré-programação gravada no chip, através dos atuadores (ex. motores).

O robô realiza diversos ciclos de percepção-ação modificando o estado do ambiente em busca da realização da tarefa.

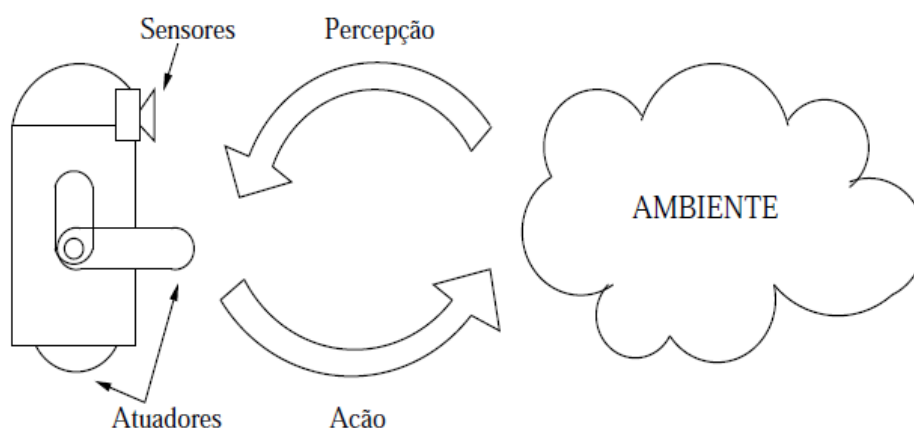


Figura 1.0 – Ciclo de interação

## 1.4 CLASSIFICAÇÃO DOS ROBÔS MÓVEIS

Diversas taxonomias têm sido utilizadas para classificar robôs móveis, contudo, não há ainda nenhuma que seja definitiva. De uma forma geral, podemos agrupar os robôs existentes de acordo com três aspectos: anatomia, tipo de controle e funcionalidade.

### Segundo a Anatomia

Quanto à anatomia, os robôs podem ser classificados em três grandes grupos: os robôs aéreos, aquáticos e terrestres.

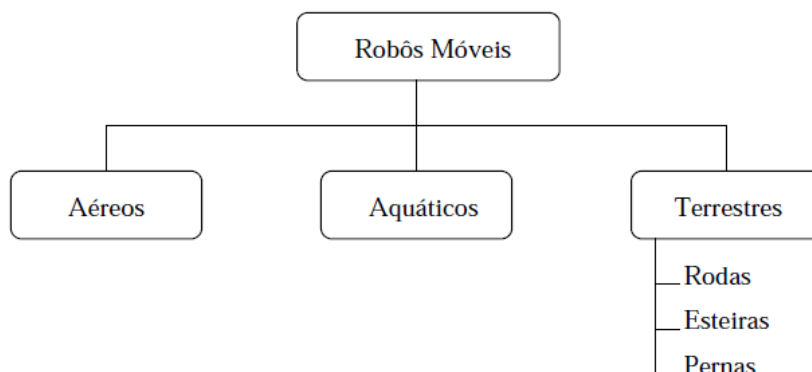


Figura 1.1 – Classificação segundo a anatomia

Os robôs terrestres são os mais populares e podem utilizar três tipos diferentes de locomoção: por rodas, por esteiras ou por pernas. Vamos enfatizar os robôs terrestres com rodas, pois estes são o nosso foco neste curso.

**Robôs com rodas:** estes são os mais simples, pois não necessitam de um hardware tão complexo quanto os robôs com esteiras e pernas, além disso, torna-se fácil transformar brinquedos em robôs. A principal desvantagem no uso de rodas é que em terrenos irregulares, o desempenho pode não ser satisfatório. Em geral, a roda do robô deve possuir raio igual ou maior que os obstáculos que ele irá transpor.

Nas figuras a seguir podemos observar exemplos de robôs com rodas e a sua colocação nas competições. Estes foram desenvolvidos por alunos do IFCE Campus Maracanaú do curso de Automação Industrial e integrantes do Grupo SanUSB, no intuito de disputar competições de robótica de desvio de obstáculos e seguidor de labirinto.

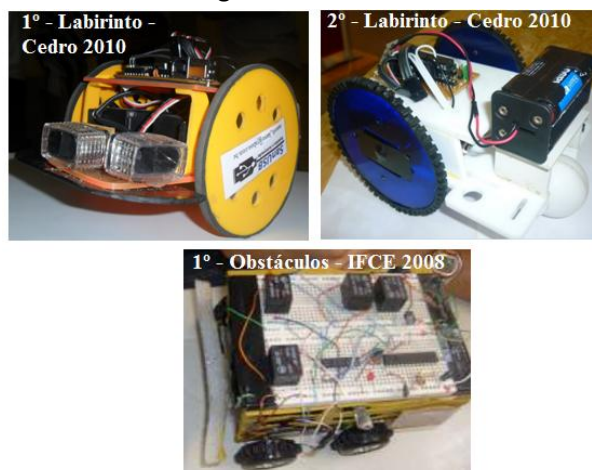


Figura 1.2 – Exemplos de robôs com rodas

## Segundo o tipo de controle

Quando classificados segundo o tipo de controle, os robôs podem ser separados em três categorias:

- **Tele-operados:** onde o operador realiza todos os movimentos que o robô deve fazer;
- **Semi-autônomos:** o operador indica o macro comando a ser executado e o robô o faz sozinho;
- **Autônomos:** o robô realiza a tarefa sozinho, tomando suas próprias decisões baseando-se nos dados obtidos do ambiente.

## Segundo a Funcionalidade

Ao agruparmos os robôs segundo a funcionalidade, percebemos 5 grupos: robôs industriais, robôs de serviço, robôs de campo, robôs pessoais e robôs educacionais. Contudo, há uma sobreposição entre os três primeiros, devido à diferença dos ambientes onde atuam e a necessidade de maior autonomia.

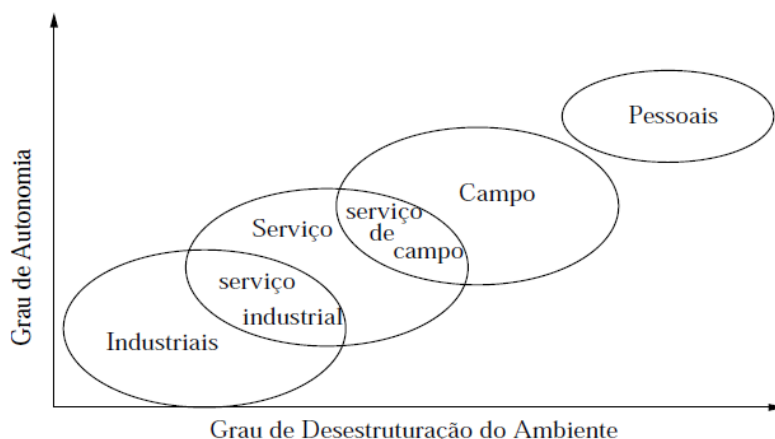


Figura 1.3 – Classificação segundo a funcionalidade

**Robôs Industriais:** são os robôs utilizados em linhas de produção. Estes robôs recebem tarefas determinadas *a priori* na forma de uma sequência explícita de ações e executam este programa automaticamente. O ambiente é completamente estruturado e ajustado para a execução da tarefa. Neste ambiente, o robô tem conhecimento exato da sua posição e da localização dos objetos. Em geral, os robôs móveis industriais são plataformas móveis utilizadas para tarefas pesadas, como transporte de materiais e produtos finais em sistemas de manufatura. Estes são denominados AGVs (*Automated Guided Vehicles*) e são programados para seguir linhas desenhadas no chão.

**Robôs de serviço industrial:** este tipo de robô possui as características de um robô de serviço, porém atua em um ambiente completamente estruturado.

**Robôs de serviço:** são os robôs móveis utilizados para serviços gerais. O ambiente é estruturado e o robô possui um modelo deste ambiente, que é conhecido previamente, porém possui certa autonomia, pois processa informação sensorial para atuar em situações imprevistas, como desviar de uma pessoa ou objeto. São utilizados em serviços de limpeza, de vigilância e no transporte de materiais leves (correspondências internas, material hospitalar, etc.).



Figura 1.4 – Robô lixeira

**Robôs de serviço de campo:** atuam em ambientes externos que podem ser previamente modelados ou não. Em geral, caso haja um modelo, este é precário e há a necessidade do processamento sensorial para complementar o modelo existente. Utilizados em tarefas agrícolas e navegação em auto-estradas.

**Robôs de campo:** trabalham em ambientes não estruturados, pouco conhecidos e geralmente perigosos. São responsáveis por exploração espacial ou de cavernas, mineração e limpeza de acidentes nucleares.

**Robôs pessoais:** são robôs vendidos em prateleiras, que não desenvolvem tarefas específicas, mas interagem com os humanos e aprendem a localizar-se no ambiente.



Figura 1.5 – Exemplos de robô pessoal: cãozinho AIBO e MIO (Sony)

**Robôs educacionais:** apresentam características semelhantes aos robôs pessoais e, além disso, são utilizados como ferramenta educacional no intuito de ajudar no ensino de disciplinas como física, matemática, informática e eletrônica. São desmontáveis e reprogramáveis de forma que na execução destas duas tarefas, o usuário aprende como funciona todo o sistema eletrônico de controle e pode alterar parâmetros de programação e de sensoriamento, observando assim o desempenho do robô mediante as ações pré-programadas. Veremos mais a frente o passo a passo da montagem de um robô móvel educacional.





Figura 1.6 – Exemplo de robô móvel educacional

## 1.5 DESAFIOS DA ROBÓTICA MÓVEL

Construir um robô móvel envolve uma série de problemas. É necessário lidar com erros de posição, de orientação, ruídos nas leituras dos sensores, problemas com a carga das baterias, erros de programação, entre outros problemas que, quando o robô passa de protótipo em laboratório, para produto funcional, encontra no ambiente; pois há variação de luminosidade, interferências de outros sistemas eletrônicos, entre outros.

Quando desenvolvemos robôs, procuramos nos espelhar em nossas habilidades; e essa busca pelo desenvolvimento de máquinas com características humanas nos apresenta uma série de desafios.

### Reconhecimento

Fazer com que o robô reconheça o ambiente no qual está inserido é bastante complexo. Deve-se trabalhar bem o processamento de imagens e dados, além das dificuldades encontradas ao tratar diferenças de luminosidade e ruído nos dados advindos dos sensores.

### Aprendizado

Ao tornar o robô mais complexo, adicionando a ele novos sensores e atuadores, estamos também acrescentando mais complexidade ao trabalho de programação de ações do robô, além de dificultar a tarefa de calibração dos sensores e coordenação dos movimentos. O desenvolvimento de algoritmos de aprendizado permite que o robô calibre seus sensores e desempenhe a sua tarefa de forma mais autônoma e adaptável.

### Cognição

O maior de todos os desafios é explorar os limites da inteligência artificial. Até que ponto podemos criar novas formas para armazenar conhecimento e utilizar este conhecimento para a tomada de decisões, dotando o robô de capacidades cognitivas de alto nível que o permitam total autonomia com relação ao meio.

*Notas:*

---

**Automação:** Qualquer sistema automático que executa operações sem a interferência do homem. Um robô pode ser considerado um sistema automático.





---

**Domótica:** Tipo de automação destinada a residências (Automação Residencial). Ex: controle temporal de cargas, sistemas de alarme em geral.

---

Para a construção de um novo conhecimento o sujeito precisa vivenciar situações em que possa relacionar, comparar, diferenciar e integrar os conhecimentos. Assim, ele estará construindo conhecimentos de longa duração, pois os incorporou em situações práticas e em experiências vivenciadas.

A sequência metodológica de práticas abordadas para construção dos robôs móveis, utilizando o software SanUSB, é abordada de acordo com o seguinte conteúdo teórico e prático:

- 1- Comparação e conceitos de sistemas robóticos, de automação e domótica;
- 2- Atuadores aplicados em robótica;
- 3- Principais motores utilizados em robótica móvel (servo-motor, motor CC em ponte H com micro-relés e motor de passo);
- 4- Sistema microcontrolado de programação e controle de robôs móveis;
- 5- Principais sensores utilizados em robótica;
- 6- Construção de robôs móveis;
- 7- Implementação de comportamentos com introdução aos princípios de desvio de obstáculos e de labirinto.

---

## 2. FERRAMENTA SanUSB

O sistema de desenvolvimento SanUSB é uma ferramenta composta de software e hardware básico da família PIC18Fxx5x com interface USB. Esta ferramenta livre se mostra eficiente no desenvolvimento rápido de projetos reais, pois não há necessidade de remover o microcontrolador para a atualização do firmware.

Além disso, esta ferramenta se mostra eficaz no ensino e na difusão de microcontroladores, bem como em projetos de eletrônica e informática, pois todos os usuários podem desenvolver projetos reais no ambiente de ensino ou na própria residência sem a necessidade de um equipamento para gravação de microcontroladores.

Dessa forma, ela é capaz de suprimir:

- 1- Um equipamento específico para gravação de um programa no microcontrolador;
- 2- Conversor TTL - EIA/RS-232 para comunicação serial bidirecional emulado através do protocolo CDC;
- 3- Fonte de alimentação, já que a alimentação do PIC provém da porta USB do PC. *É importante salientar que cargas indutivas como motores de passo ou com corrente acima de 400mA devem ser alimentadas por uma fonte de alimentação externa.*
- 4- Conversor analógico-digital (AD) externo, tendo em vista que ele dispõe internamente de **10** ADs de 10 bits;

- 5- *Software* de simulação, considerando que a simulação do programa e do *hardware* podem ser feitas de forma rápida e eficaz no próprio circuito de desenvolvimento ou com um *protoboard* auxiliar.

Utilizaremos esta ferramenta como circuito de controle do robô móvel que aprenderemos a desenvolver neste curso.

Além de todas estas vantagens, os *laptops* e alguns computadores atuais não apresentam mais interface de comunicação paralela e nem serial EIA/RS-232, somente USB.

Como pode ser visto, esta ferramenta possibilita que a compilação, a gravação e simulação real de um programa, como também a comunicação serial através da emulação de uma porta COM virtual, possam ser feitos de forma rápida e eficaz a partir do momento em o microcontrolador esteja conectado diretamente a um computador através de um cabo USB.

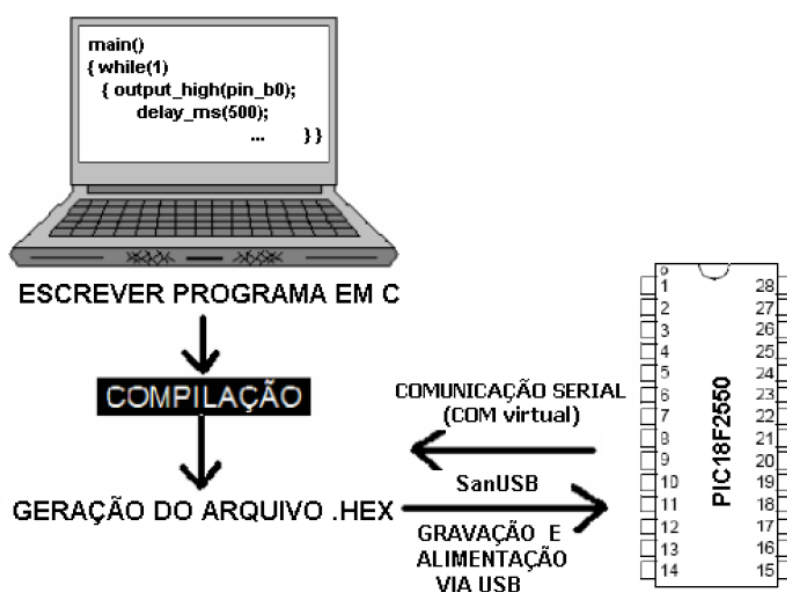


Figura 2.0 – Esquemático do sistema de gravação

## 2.1 UTILIZANDO A FERRAMENTA SanUSB

A transferência de programas para os microcontroladores é normalmente efetuada através de um hardware de gravação específico. Através desta ferramenta, é possível efetuar a descarga de programas para o microcontrolador diretamente de uma porta USB de qualquer PC, bem como gerar uma porta de comunicação serial virtual utilizando qualquer microcontrolador PIC com interface USB (18F2550, 18F2455, 18F4550, 18F4455, etc.).

Para que todas essas funcionalidades sejam possíveis, é necessário gravar, anteriormente e somente uma vez, com um gravador específico para PIC, o gerenciador de gravação pela USB GerenciadorLinux.hex disponível na pasta completa da ferramenta no link abaixo, onde também é possível baixar periodicamente as atualizações dessa ferramenta e a inclusão de novos programas: <http://www.4shared.com/file/sIzWBP4r/100727SanUSB.html>

Para que os programas em C possam ser gravados no microcontrolador via USB, é necessário compilá-los, ou seja, transformá-los em linguagem de máquina hexadecimal (.hex). Existem diversos compiladores que podem ser utilizados por esta ferramenta, entre eles o SDCC, o C18, o Hi-Tech e o CCS.

Devido à didática das funções e bibliotecas disponíveis para emulação serial, diversos periféricos e *multitasking*, o compilador mais utilizado nos projetos é o CCS. Uma versão demo funcional do compilador utilizado com bibliotecas de suporte a USB é possível baixar em: <http://www.4shared.com/file/Mo6sQJs2/100511Compilador.html>

Para instalar o CCS também no Linux através do Wine basta seguir as seguintes instruções da seguinte vídeo-aula: <http://www.youtube.com/watch?v=5-kigedbfvg>. Caso grave o novo gerenciador de gravação pela USB "GerenciadorLinux.hex", não esqueça de colar o novo arquivo cabeçalho SanUSB.h e as outras bibliotecas disponível na pasta ExemploBibliotecas para a pasta Drivers localizada dentro da pasta instalada do compilador ( C:\Arquivos de programas\PICC\Drivers ).

A representação básica do circuito SanUSB montado em protoboard com um microcontrolador de 28 pinos é mostrada a seguir:

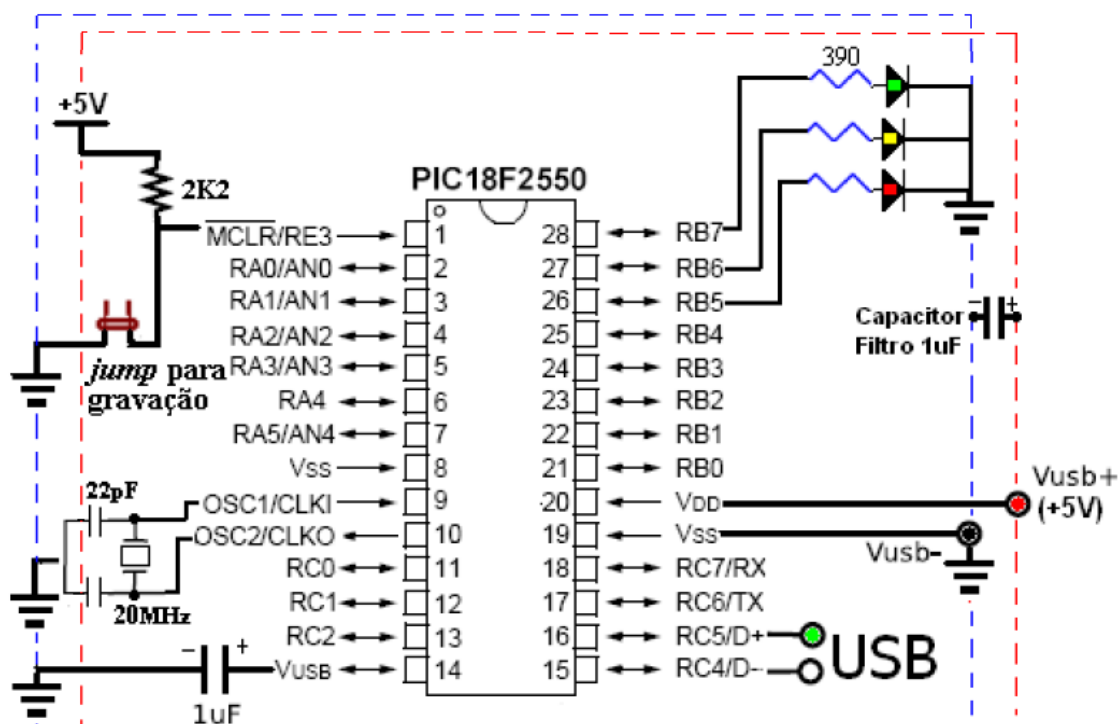


Figura 2.1 – Esquemático de montagem do circuito com PIC 18F2550

Para um microcontrolador de 40 pinos, o circuito é mostrado abaixo:

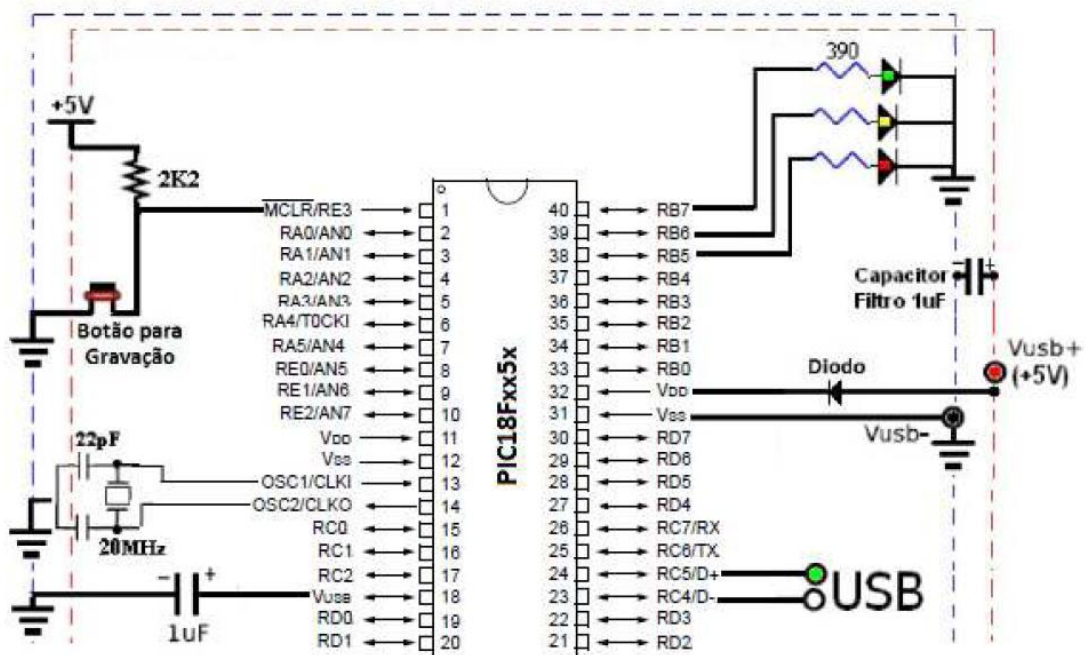


Figura 2.2 - Esquemático de montagem do circuito com PIC 18F4x5x

Os componentes básicos do circuito são:

- 1 microcontrolador da família PIC USB (18F2550, 18F2455, 18F4550, etc.);
- 1 cristal de 20MHz;
- 2 capacitores de 22pF;
- 2 capacitores de 1uF (um no pino 14 Vusb e outro entre o +5V e o Gnd) ;
- 3 leds e 3 resistores de 390 (só é necessário um led com resistor no pino B7);
- 1 resistor de 2k2 e um botão ou fio para gravação no pino 1;
- 1 diodo qualquer entre o +5V e o o pino Vdd;
- 1 Cabo USB qualquer.

Note que, ao conectar o cabo USB e alimentar o microcontrolador, com o pino 1 no Gnd (0V), através do botão ou de um simples fio, é Estado para Gravação via USB (led no pino B7 aceso) e que, após o reset com o pino 1 no Vcc (+5V através do resistor fixo de 2K2 sem o jump) é Estado para Operação do programa aplicativo (firmware) que foi compilado.

O cabo USB apresenta normalmente quatro fios, que são conectados ao circuito do microcontrolador nos pontos mostrados na figura acima, onde normalmente, o fio Vcc (+5V) do cabo USB é vermelho, o Gnd (Vusb-) é marron ou preto, o D+ é azul ou verde e o D- é amarelo ou branco.

Note que a fonte de alimentação do microcontrolador nos pinos 19 e 20 e dos barramentos vermelho (+5V) e azul (Gnd) do circuito provêm da própria porta USB do computador. Para ligar o cabo USB no circuito é possível cortá-lo e conectá-lo direto na protoboard, com fios rígidos soldados, como também é possível conectar sem cortá-lo, em um protoboard ou numa placa de circuito impresso, utilizando um conector USB fêmea. O diodo de proteção colocado no pino 20 entre o Vcc da USB e a alimentação do microcontrolador serve para proteger contra corrente reversa caso a tensão da porta USB esteja polarizada de forma inversa.

A figura abaixo mostra a ferramenta SanUSB montada em protoboard seguindo o circuito anterior e a posição de cada terminal no conector USB a ser ligado no PC. Cada terminal é conectado diretamente nos pinos do microcontrolador pelos quatro fios correspondentes do cabo USB.

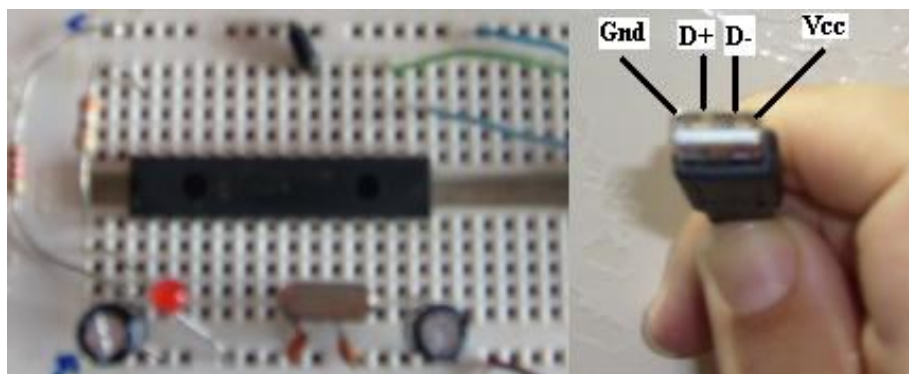


Figura 2.3 – Montagem do circuito com PIC em protoboard e sequência da USB

É importante salientar que, para o perfeito funcionamento da gravação via USB, o circuito desta ferramenta deve conter um capacitor de filtro entre 0,1uF e 1uF na alimentação que vem da USB, ou seja, colocado entre os pinos 20 (+5V) e 19 (Gnd).

Caso o sistema microcontrolado seja embarcado como, por exemplo, um robô, um sistema de aquisição de dados ou um controle de acesso, ele necessita de uma fonte de alimentação externa, que pode ser uma bateria comum de 9V ou um carregador de celular. A figura abaixo mostra o PCB, disponível nos Arquivos do Grupo SanUSB, e o circuito para esta ferramenta com entrada para fonte de alimentação externa. Para quem deseja obter o sistema pronto para um aprendizado mais rápido, é possível também encomendar placas de circuito impresso da ferramenta SanUSB, como a foto da placa abaixo, entrando em contato com o grupo SanUSB através do e-mail: [sanusb\\_laese@yahoo.com.br](mailto:sanusb_laese@yahoo.com.br).



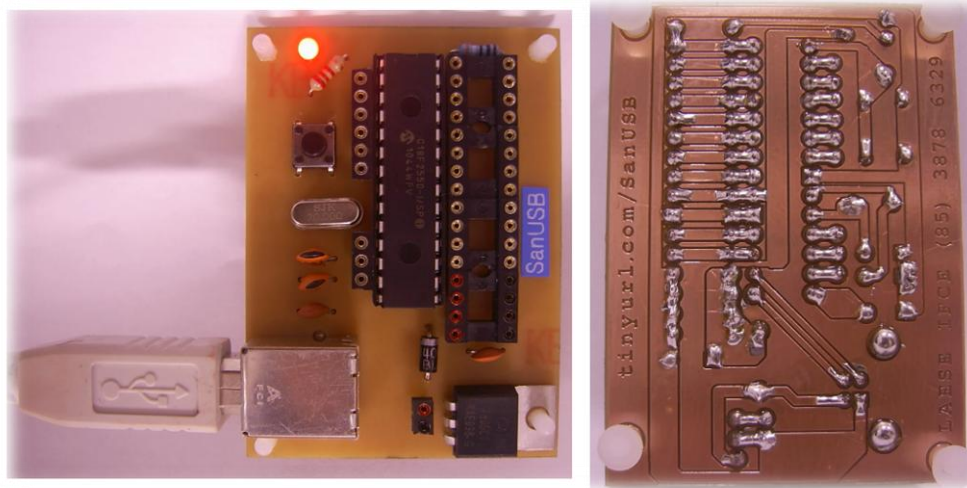


Figura 2.4 – PCI (Placa de Circuito Impresso) da ferramenta SanUSB

Se preferir confeccionar a placa, basta seguir o circuito abaixo:

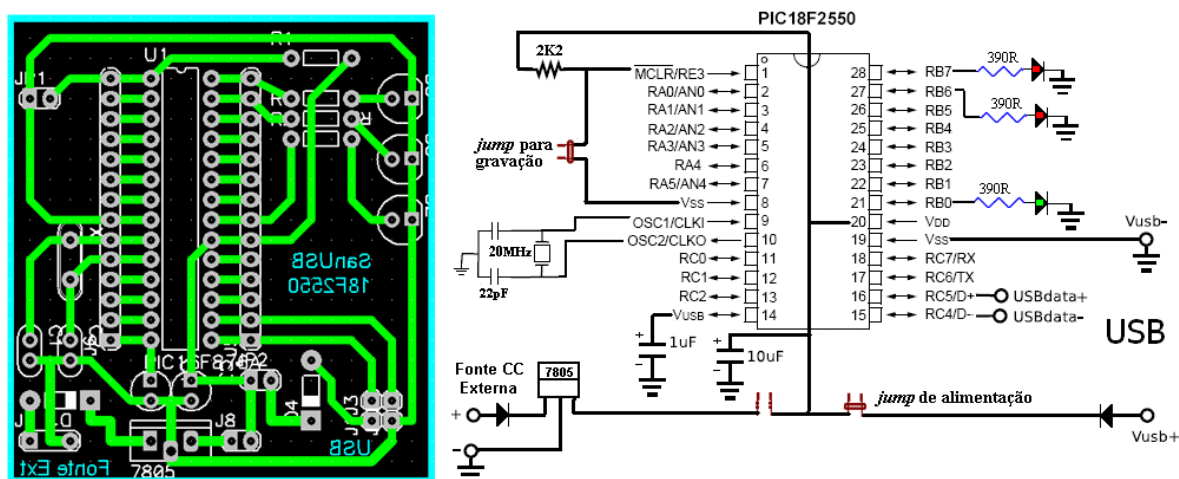


Figura 2.5 – Esquema para construção de PCI da ferramenta SanUSB

Para obter vários programas-fonte e vídeos deste sistema livre de gravação, comunicação e alimentação via USB, basta se cadastrar no grupo de acesso livre [www.tinyurl.com/SanUSB](http://www.tinyurl.com/SanUSB) e clicar no item Arquivos.

Durante a programação do microcontrolador basta inserir, no início do programa em C, a biblioteca cabeçalho SanUSB (#include <SanUSB.h>) contida dentro da pasta SanUSB\_User\Exemplos\_SanUSB e que você já adicionou dentro da Drivers localizada na pasta instalada do compilador ( C:\Arquivos de programas\PICC\Drivers ). Essa biblioteca contém instruções do PIC18F2550 para o sistema operacional, configurações de fusíveis e habilitação do sistema Dual Clock, ou seja, oscilador RC interno de 4 MHz para CPU e cristal oscilador externo de 20 MHz para gerar a frequência de 48MHz da comunicação USB, através de prescaler multiplicador de frequência.

Como a frequência do oscilador interno é de 4 MHz, cada incremento dos temporizadores corresponde a um microssegundo. O programa “*exemplo1.hex*” a seguir comuta um *led* conectado no pino B7 a cada 0,5 segundo.

```
#include <SanUSB.h>

void main()
{
  clock_int_4MHz();//Função necessária para habilitar o dual clock (48MHz para USB e 4MHz para CPU)

  while (1)
  {
    output_toggle(pin_B7); // comuta Led na função principal
    delay_ms(500);
  }
}
```

O programa “*pisca3.hex*” abaixo pisca três *leds* conectados nos pinos B5, B6 e B7.

```
#include <SanUSB.h>

main(){
  clock_int_4MHz();//Função necessária para habilitar o dual clock (48MHz para USB e 4MHz para CPU)

  while (1)
  {
    output_high(pin_B5); // Pisca Led na função principal
    delay_ms(500);
    output_low(pin_B5);
    output_high(pin_B6);
    delay_ms(500);
    output_low(pin_B6);
    output_high(pin_B7);
    delay_ms(500);
    output_low(pin_B7);
  }
}
```

Os arquivos compilados.hex assim como os firmwares estão disponíveis na pasta 100727SanUSB disponível nos arquivos do grupo SanUSB: [www.tinyurl.com/SanUSB](http://www.tinyurl.com/SanUSB) .


## 2.2 GRAVANDO O MICROCONTROLADOR VIA USB NO WINDOWS

Para executar a gravação com a ferramenta *SanUSB*, é importante seguir os seguintes passos:


1. Copie, para um diretório raiz C ou D, a pasta SanUSB obtida no grupo [www.tinyurl.com/SanUSB](http://www.tinyurl.com/SanUSB).



2. Grave no microcontrolador, somente uma vez, com um gravador específico para PIC, o novo gerenciador de gravação pela USB “*GerenciadorWin.hex*” disponível na pasta SanUSB.
3. Pressione o botão ou conecte o *Jump* de gravação do pino 1 do circuito SanUSB no Gnd para a transferência de programa do PC para o microcontrolador.
4. Conecte o cabo USB entre o PIC e o PC. Se o circuito SanUSB estiver correto acenderá o *led* do pino B7.

Se for o **Windows 7**, vá em propriedades do sistema -> Configurações avançadas do sistema -> Hardware -> Gerenciador de dispositivos e clique com botão direito no driver USB do microcontrolador e atualizar Driver, apontando para a pasta DriverWindows7SanUSB. Após a instalação no Windows 7, clique com o botão direito sobre o ícone de gravação SanUSB , selecione Propriedades, Compatibilidade e escolher Windows Vista (Service Pack 1).

Se for o **Windows XP**, o windows irá perguntar, somente a primeira vez, onde está o *Driver* de instalação, então escolha a opção *Instalar de uma lista ou local específico (avanzado)*. Após Avançar, escolha a opção Incluir este local na pesquisa e selecione a pasta “DriverWinXPSanUSB”, onde está o *driver* “sanusb\_device”. Durante a instalação, o Windows abrirá uma janela sobre a instalação, selecione a opção continuar assim mesmo e o *driver* será instalado.

5. Abra o aplicativo **SanUSB** . Se estiver conectado corretamente, o *led* conectado no pino B7 acende e aparecerá a seguinte tela:

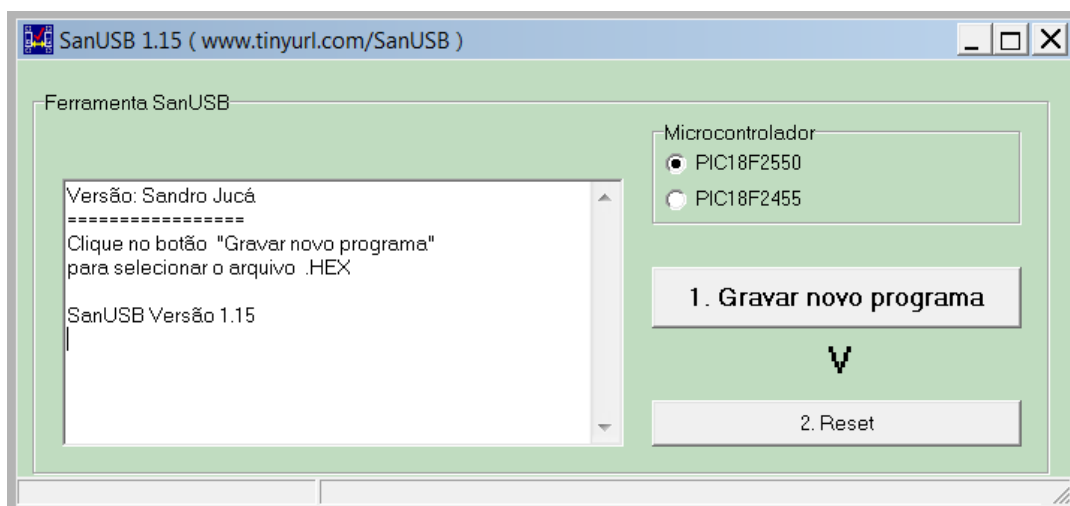


Figura 2.6 – Tela do software de gravação do programa em linguagem C no PIC - Windows

6. Clique em *1. Gravar novo programa* e escolha o programa *.hex* que deseja gravar, como por exemplo, o programa compilado *pisca.hex* da pasta Exemplos\_SanUSB. Este programa pisca 3 *leds* conectados nos pinos B5, B6 e B7;
7. Após a gravação do programa, retire o *jump* do pino de gravação e clique em *Reset*. Pronto o programa estará em operação.

Para programar novamente, basta colocar o *jump* de gravação, retire o *jump* de alimentação, coloque-o novamente e repita os passos anteriores a partir do passo 4.

Se a nova programação não funcionar, retire o conector USB do computador e repita os passos anteriores a partir do passo 3.

### 2.3 GRAVANDO O MICROCONTROLADOR VIA USB NO LINUX

Esta aplicação substitui a gravação via USB pelo terminal do Linux, pois é uma forma mais simples e direta de gravação. Com apenas dois cliques no instalador automático “SanUSB.deb” é possível instalar este aplicativo em qualquer máquina com Linux (Ubuntu 10.04, equivalente ou posterior).

Depois de instalado, a interface de gravação é localizada em Aplicativos -> acessórios. Se você já tem o Java instalado (JRE ou SDK) baixe o instalador “automático.deb” atualizado disponível no link: [http://www.4shared.com/file/RN4xpF\\_T/sanusb\\_Linux.html](http://www.4shared.com/file/RN4xpF_T/sanusb_Linux.html) contido também na pasta que se encontra neste link <http://www.4shared.com/file/sIZwBP4r/100727SanUSB.html>.

Se ainda não tem o Java (JRE ou SDK), baixe o instalador SanUSB configurado com o Java JRE disponível em: <http://www.4shared.com/file/3mhWZS5g/sanusb.html>. A figura abaixo mostra a interface gráfica desenvolvida para gravação direta de microcontroladores via USB:

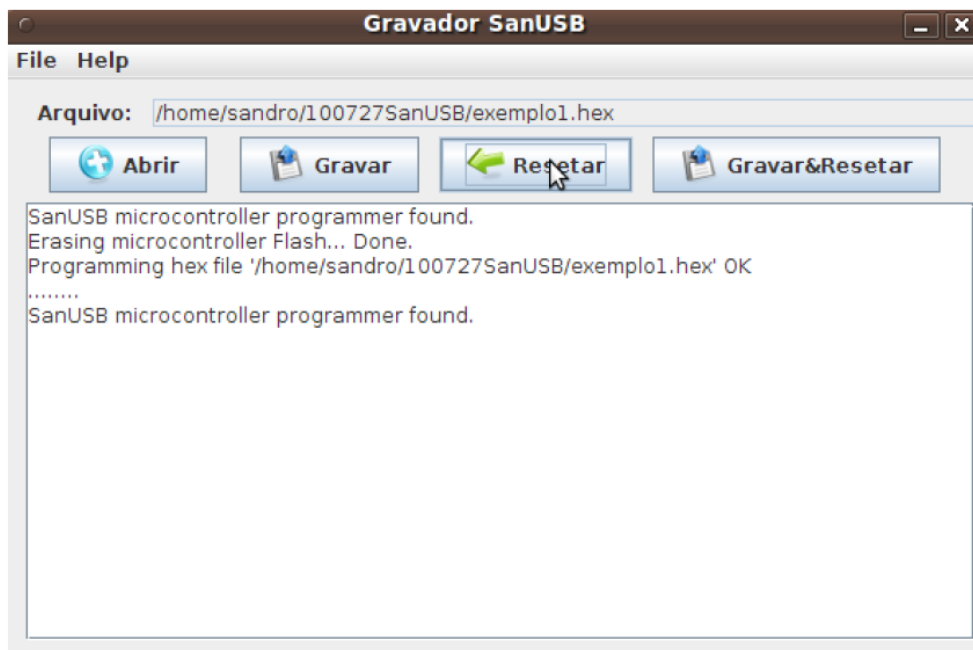


Figura 2.7 – Tela do software de gravação do programa em linguagem C no PIC - Linux

Neste aplicativo, estão disponíveis botões para **Abrir** o programa em hexadecimal compilado, para **Gravar** o programa hexadecimal no microcontrolador via USB e para **Resetar** o microcontrolador no intuito de colocá-lo em operação. A interface apresenta ainda um botão para **gravar e resetar** automaticamente.

É importante salientar que para utilizar esta ferramenta no Linux é necessário estar *logado* com permissão para acessar a porta USB como, por exemplo, super-usuário (*sudo su*), e que para estabelecer comunicação com o microcontrolador é necessário gravar anteriormente no microcontrolador, somente uma vez, com qualquer gravador específico para PIC, o gerenciador de gravação pela USB “GerenciadorLinux.hex”, disponível na pasta SanUSB ou em <http://www.4shared.com/file/HGrf9nDz/Gerenciador.html>.

Após gravar o “GerenciadorLinux.hex” com um gravador convencional para PIC, coloque o circuito SanUSB em modo de gravação pela USB (pino 1 ligado ao Gnd (0V) através de botão, jump ou fio ) e conecte o cabo USB do circuito no PC. Se o circuito SanUSB estiver correto, acenderá o led do pino B7. Pronto, o sistema já está preparado para gravar no microcontrolador, de forma simples e direta, quantos programas.hex você desejar utilizando a interface USB. Para programar novamente, basta pressionar o botão de gravação no pino 1, conectar o jump ou fio e em seguida desconectar e conectar o cabo USB de alimentação, selecionar o programa.hex desejado em **Abrir** e pressione o botão **Gravar&Resetar**.

### 3. ATUADORES APLICADOS EM ROBÓTICA

Os motores mais utilizados com sistemas microcontrolados são os motores CC, motores de passo e servo-motores. A figura abaixo mostra a disposição dos transistores de potência para atuação bidirecional de motores de passo e motores CC.

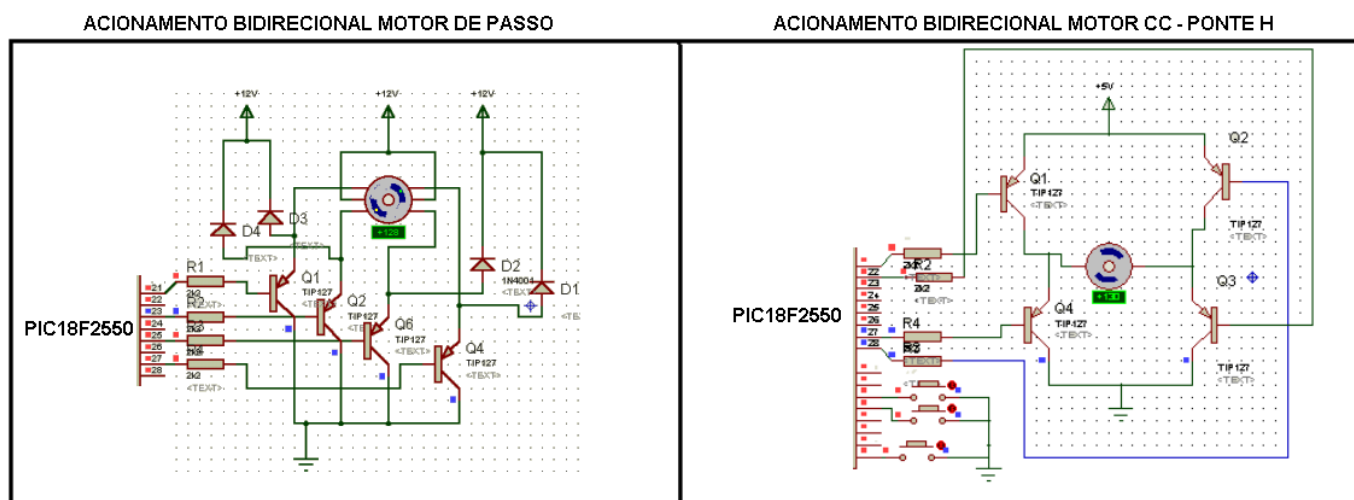


Figura 3.0 – Acionamento Bidirecional de motores

### 3.1 ACIONAMENTO DE MOTORES CC (CORRENTE CONTÍNUA) DE BAIXA TENSÃO

#### Motores de equipamentos eletrônicos

São abundantes no mercado em função da ampla gama de utilização, conseqüentemente, existem em várias dimensões, tensões, pesos, características e são fáceis de encontrar em sucatas como videocassete, brinquedos, impressoras, etc.



Figura 3.1 – Exemplos de motores CC

#### Motores elétricos utilizados em automóveis

Os motores utilizados em automóveis são todos com tensão nominal a 12 volts, são robustos e normalmente projetados para condições extremas, tais como poeira, calor, variações de tensão e corrente, entre outros. Algumas opções são ideais para aplicação no robô por serem compactos, fortes, alta rotação e leves, além de serem muito fáceis de conseguir em oficinas e empresas do ramo. Os motores mais usados em projetos em geral são de trava elétrica das portas, bomba do limpador de pára-brisa e de gasolina, bomba de combustível, motor do vidro-elétrico, motor da ventoinha, motor do ventilador interno, limpador de pára-brisa dianteiro e traseiro, bomba hidráulica do freio ABS, de carros de brinquedo.



Figura 3.2 – Exemplos de motores CC de automóveis

Além do acionamento elétrico, os motores CC de baixa potência utilizados em automação e robótica, apresentam geralmente uma caixa de redução, que é um equipamento composto por engrenagens, com o intuito de reduzir a velocidade de rotação do eixo (ou angular) e aumentar o torque do motor. O torque varia em função da força aplicada (F) e do raio de giro (nas engrenagens é a metade do diâmetro primitivo), segundo a equação  $T = F \times r$ .

Sendo: F = força (em Newtons), r = raio de giro (em metros) e T = torque (em N.m).

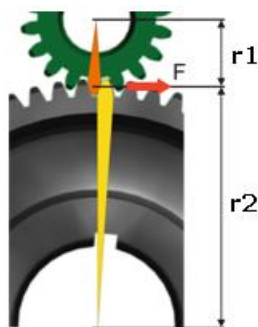


Figura 3.3 – Comparação entre os raios de engrenagens

Já que o motor imprime uma força constante, a variação do torque entre engrenagens ocorre devido ao raio de giro. Na prática em um sistema de engrenagens, comprovada pelas equações abaixo, quanto maior o diâmetro da engrenagem movida ( $D_2$ ), maior o torque ( $T_2$ ) proporcional e menor a velocidade de rotação ( $n_2$ ). Considerando a engrenagem 1 com motora e a engrenagem 2 como movida, tem-se:

$$F_{const} \rightarrow T_1/r_1 = T_2/r_2 \rightarrow T_1/D_1 = T_2/D_2$$

$$T_2 \cdot D_1 = T_1 \cdot D_2$$

$$n_2 \cdot D_2 = n_1 \cdot d_1$$

### Coroa e o parafuso com rosca sem-fim

A coroa e o parafuso com rosca sem-fim compõem um sistema de transmissão muito utilizado principalmente nos casos em que é necessária elevada redução de velocidade ou um elevado aumento de força, como nos redutores de velocidade.

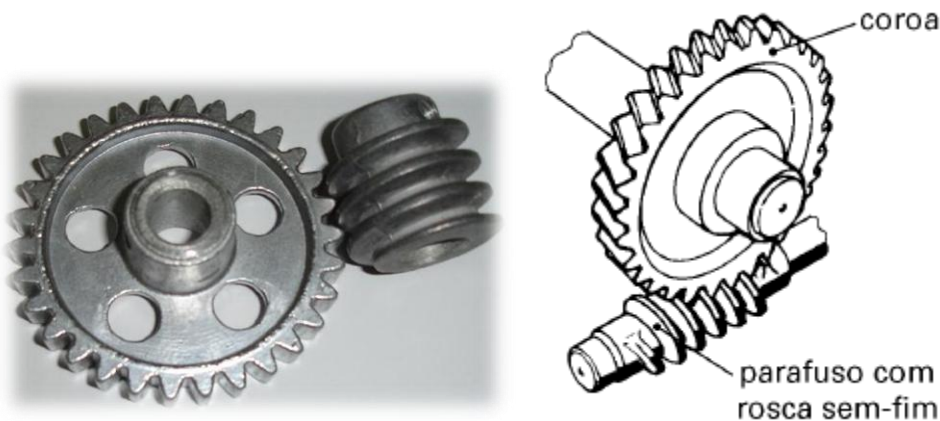


Figura 3.4 – Parafuso com rosca sem-fim

O número de entradas do parafuso tem influência no sistema de transmissão. Se um parafuso com rosca sem-fim tem apenas uma entrada (mais comum) e está acoplado a uma coroa de 60 dentes, em cada volta dada no parafuso a coroa vai girar apenas um dente. Como a coroa tem 60 dentes, será necessário dar 60 voltas no parafuso para que a coroa gire uma volta. Assim, a rpm da coroa é 60 vezes menor que a do parafuso. Se, por exemplo, o parafuso com rosca sem-fim está girando a 1.800 rpm, a coroa girará a 1.800 rpm, divididas por 60, que resultará em 30 rpm.

Suponhamos, agora, que o parafuso com rosca sem-fim tenha duas entradas e a coroa tenha 60 dentes. Assim, a cada volta dada no parafuso com rosca sem-fim, a coroa girará dois dentes. Portanto, será necessário dar 30 voltas no parafuso para que a coroa gire uma volta.

Assim, a rpm da coroa é 30 vezes menor que a rpm do parafuso com rosca sem-fim. Se, por exemplo, o parafuso com rosca sem-fim está girando a 1.800 rpm, a coroa girará a 1.800 divididas por 30, que resultará em 60 rpm. A rpm da coroa pode ser expressa pela equação:

$$i = N_c \cdot Z_c = N_p \cdot Z_p$$

$$N_c = (N_p \cdot Z_p) / Z_c$$

Onde:

**N<sub>c</sub>** = número de rotações da coroa (rpm)

**Z<sub>c</sub>** = número de dentes da coroa

**N<sub>p</sub>** = número de rotações do parafuso com rosca sem-fim (rpm)

**Z<sub>p</sub>** = número de entradas do parafuso

## AS POSSIBILIDADES MAIS COMUNS DE CONTROLE DIGITAL DE MOTORES CC SÃO:

### Chaveamento de motores cc com transistores mosfet

Os transistores de efeito de campo MOSFET executam o chaveamento por tensão na base e podem ser utilizados no lugar dos transistores Darlington para acionamento de dispositivos de média potência, devido principalmente à menor queda de tensão e à menor dissipação em forma de calor.

Os MOSFETs apresentam alta taxa de velocidade no chaveamento e uma resistência interna muito baixa (décimos de ohms). Dessa forma, a queda de tensão nesse transistor é muito baixa, o que não acontece com transistores Darlington. A figura abaixo mostra a configuração dos pinos de um MOSFET, onde o pino 1 é o *Gate* (base), o pino 2 é o *Drain* e o pino 3 é o *Source*.

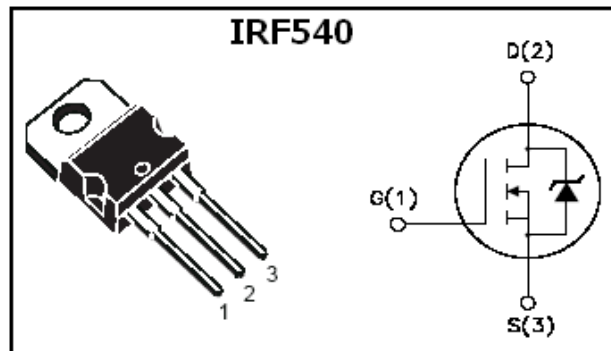


Figura 3.5 – Transistor Mosfet IRF540

O modelo IRF540 suporta tensões de chaveamento entre o *drain* e o *source* de 100V e corrente de até 22A. A figura abaixo mostra o circuito com MOSFET para acionamento de quatro motores com MOSFETs. A etapa de potência é composta pelos MOSFETs IRF530 e diodos de roda livre para proteção de tensão reversa. O funcionamento é simples. Quando o microcontrolador coloca na saída das portas de controle um '1' lógico, os transistores MOSFET entram em condução e uma tensão de 12V é aplicada sobre as cargas. Os resistores e LEDs servem somente para visualização do chaveamento. Note que o pino *Source* do MOSFET é conectado ao Gnd do microcontrolador para gerar a tensão de chaveamento no *gate* ( $V_G$ ).

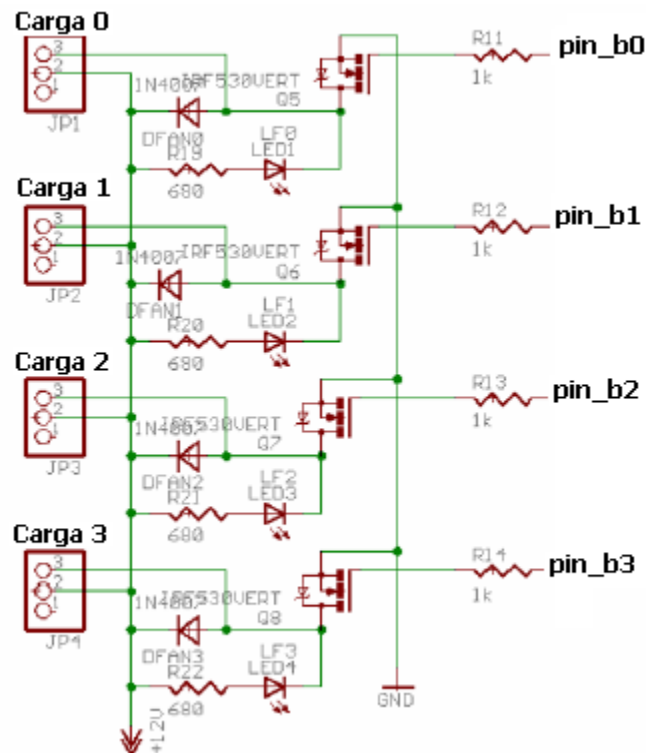


Figura 3.6 – Esquemático de conexão com o IRF540

A figura abaixo mostra o acionamento de um motor CC de 3V utilizado em robótica móvel através de um mosfet IRF540.



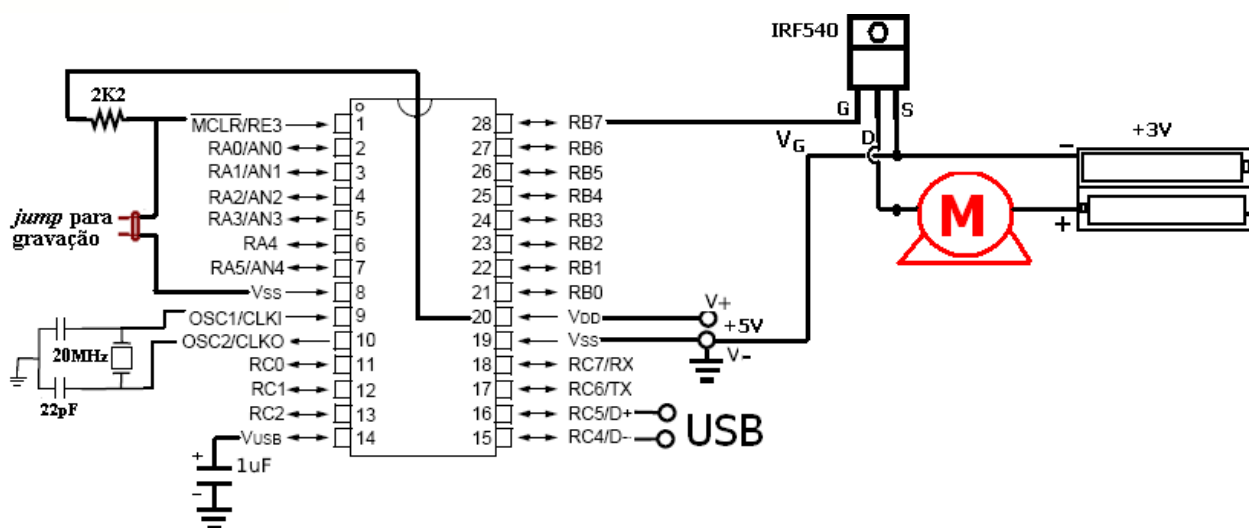


Figura 3.7 – Esquemático de montagem do IRF540 com PIC para acionamento de motor CC 3V

### Exemplo: seguidor óptico de labirinto

Neste caso é interessante definir que no princípio seguidor de parede o robô considera o obstáculo como referência a qual ele vai seguir em movimento de avanço normal. Nesse exemplo ele “cola” do lado esquerdo, ou seja, o motor direito deve ser mais rápido que o motor esquerdo, quando os dois estiverem acionados.

Seguindo as paredes do labirinto até final encontram-se quatro situações:

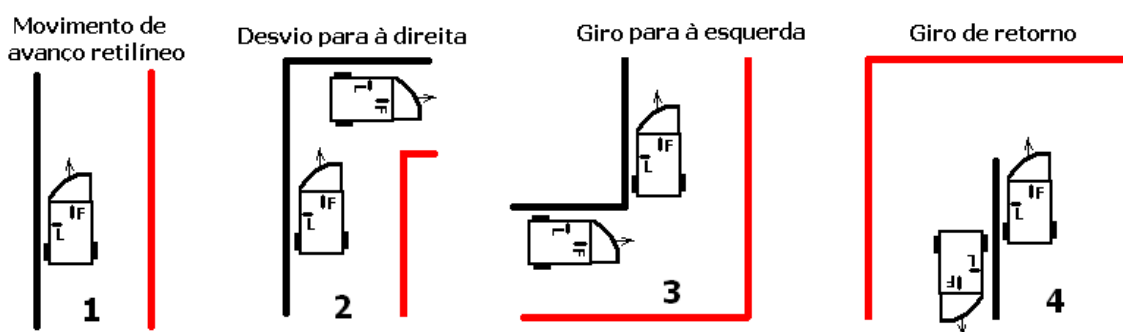


Figura 3.8 – Situações a serem contornadas por um robô em um labirinto

É importante salientar que no princípio de seguir o obstáculo, o robô não pode tocar no outro lado, em vermelho, pois ele pode tomá-lo como referência e voltar, o que o fará sair por onde entrou. Mais detalhes: <http://www.youtube.com/watch?v=QRDrG2iEFpM>

### Estabilidade do controle de movimento

Para garantir estabilidade do controle de movimento, ou seja, garantir que o robô está seguindo corretamente a referência (o obstáculo), o sensor óptico lateral (L), com sinal analógico, deve ser lido frequentemente e estar com valores que garantam a presença do obstáculo.

Caso seja acusada a ausência de obstáculo, o microcontrolador deve parar o motor esquerdo e acionar o motor direito, um determinado tempo, suficiente para garantir as situações 3 e 4. Note que o tempo de 4 é maior que o de 3, mas com o tempo de 4 na situação 3, o robô vai ser seguro pelo obstáculo até acabar o tempo de desvio e seguir em frente até encontrar obstáculo frontal ou lateral.

Caso o sensor frontal verifique obstáculo, mostrado na situação 2, o microcontrolador para o motor direito, aciona o motor esquerdo e segue em frente até encontrar obstáculo lateral, o que garante a estabilidade do movimento. Note que se o desvio para a direita for pouco, o guia oval frontal do robô conduzirá o robô à estabilidade ao tocar no obstáculo com o avanço frontal. O circuito é mostrado a seguir:

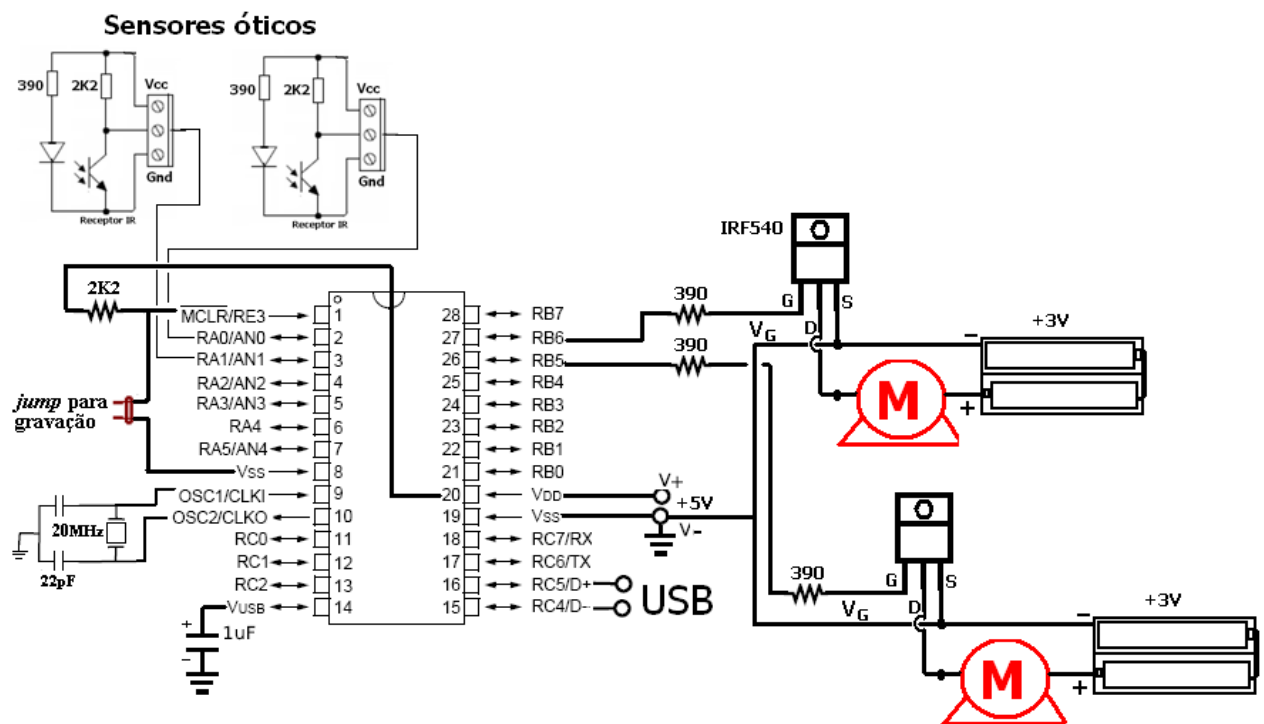


Figura 3.9 – Esquemático de montagem do circuito de controle de robô seguidor de labirinto

**Programa em linguagem C:**

```
#include <SanUSB.h> //Leitura de tensão em mV com variação de um potenciômetro

#define esquerdo pin_b6
#define direito pin_b5

int32 tensaofrente, tensaolateral, aproximacao=4790; //int32 devido aos cálculos intermediários
short int ledpisca;
unsigned int flagnaosensor=0, flagfrente=0, flaggiro=1;

main() {
clock_int_4MHz();
```



```

setup_adc_ports(AN0_TO_AN1); //Habilita entrada analógica - A0
setup_adc(ADC_CLOCK_INTERNAL);

while(1){
output_high(esquerdo);output_high(direito); //Os dois motores em frente
//*****
set_adc_channel(1);
delay_ms(10);
tensaolateral= (5000*(int32)read_adc())/1023;
if (tensaolateral<=4790) { flagnaosensor=0; flagfrente=0; flaggiro=0; } // Estabilizou, Habilita o giro e a
virada de //frente para a direita
//*****

if (flagnaosensor>=3 && flaggiro==0) {flagnaosensor=0; flaggiro=1; // Sem barreira (flagnaosensor>=3)
gire 170 ///graus
                output_low(esquerdo);output_low(direito);
output_high(pin_b7);delay_ms(500);//para
output_low(esquerdo);output_high(direito); output_low(pin_b7);delay_ms(2000); //só gira de novo se
tiver //estabilizado na lateral (flaggiro==0)

                output_high(esquerdo);output_high(direito); // Segue em frente
                while (tensaofrente>4790 &&  tensaolateral>4790) //Espera até encontrar barreira
//frontal ou lateral
                {set_adc_channel(0);
                delay_ms(10);
                tensaofrente= (5000*(int32)read_adc())/1023;
                set_adc_channel(1);
                delay_ms(10);
                tensaolateral= (5000*(int32)read_adc())/1023;
                }}
//*****
                //ANALÓGICO      DIGITAL(10 bits)
set_adc_channel(0);           // 5000 mV      1023
delay_ms(10);                // tensao      read_adc()
tensaofrente= (5000*(int32)read_adc())/1023;
//*****
if (tensaofrente<=4790 && flagfrente==0) { flagnaosensor=0; flagfrente=1; //encontrou barreira frontal
                output_low(direito);delay_ms(500);//vira para a direita

                output_high(esquerdo);output_high(direito); // Segue em frente até estabilizar

                while (tensaolateral>4790) //Espera até estabilizar a lateral
                {set_adc_channel(1); //Colado no canto frontal e lateral também estabiliza
                delay_ms(10);
                tensaolateral= (5000*(int32)read_adc())/1023;
                }}

++flagnaosensor;
output_high(esquerdo);output_high(direito); //Os dois motores em frente
ledpisca=!ledpisca;
output_bit(pin_b7,ledpisca);
delay_ms(20);
                }}

```

O fluxograma do programa do microcontrolador é mostrado a seguir:

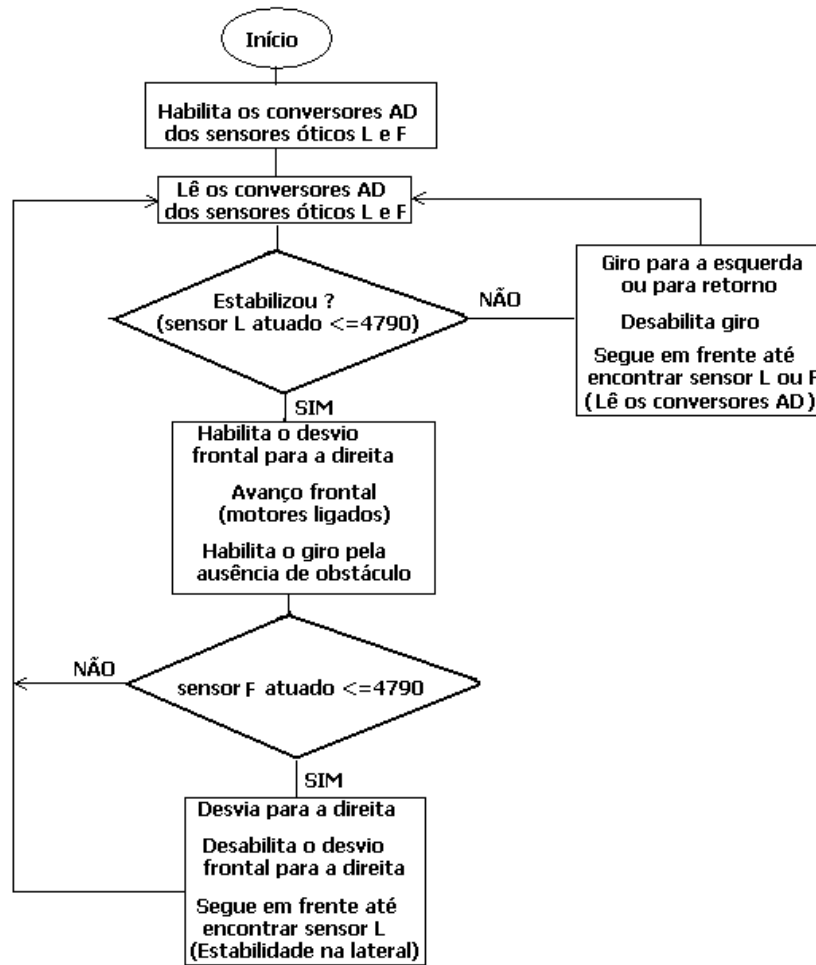


Figura 3.10 – Fluxograma referente ao programa do robô seguidor de labirinto

### Controle PWM de velocidade de um MOTOR CC

A finalidade deste controle de velocidade com a modulação de tensão por largura de pulsos (PWM) é realizar uma conversão digital-analógica que acontece devido à impedância inerente do circuito em alta frequência.

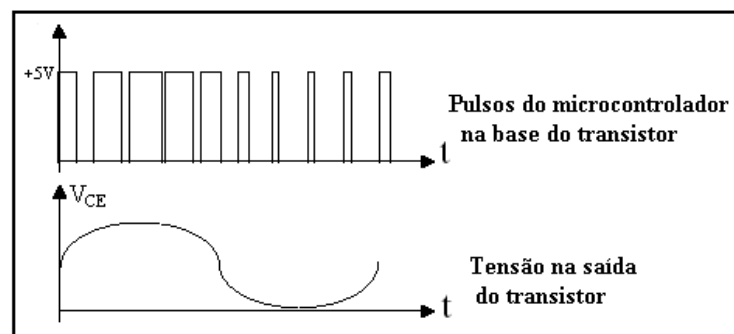


Figura 3.11 – Gráficos de ondas de tensão

A geração do PWM é produzida, geralmente, pelo chaveamento de uma tensão com amplitude constante (+5V por exemplo) em um circuito transistorizado, tendo em vista que quanto menor a largura dos pulsos emitidos na base de um transistor, menor é a saturação do transistor e, conseqüentemente, menor a tensão resultante do chaveamento.

O período  $T_0$  é o intervalo de tempo que se registra o Período repetitivo do pulso e o  $\tau_0$  é o ciclo de trabalho (*duty-cycle*), ou seja, o tempo em que o pulso permanece com a amplitude em nível lógico alto.

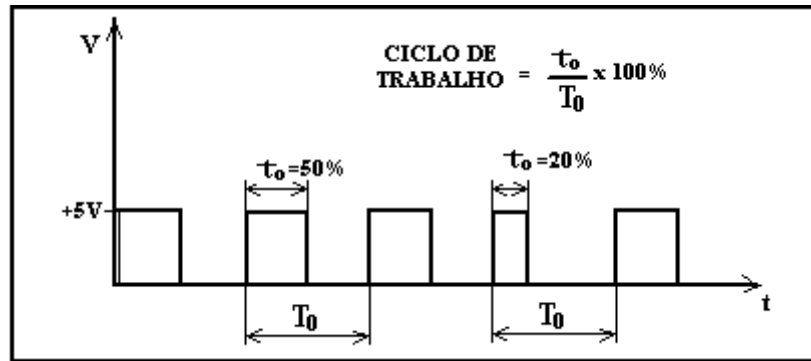


Figura 3.12 – Pulsos PWM

O programa abaixo mostra o controle de velocidade de um motor CC por PWM com período constante de 20ms, onde cada incremento ou decréscimo da onda quadrada corresponde a 1ms, ou seja, ou seja, um acréscimo ou decréscimo de 5% no ciclo de trabalho.

```
#include <SanUSB.h>

#define motor pin_b7

#define led pin_b0

unsigned int ton,toff,incton,inctoff,guardaton,guardatoff;

int1 flag1, flag2;

main() {

clock_int_4MHz();

port_b_pullups(true);

incton=2; inctoff=18; //Período de 20ms - Passo mínimo de tempo = 1ms (5% (1/20) do duty cycle )

guardaton=read_eeprom(10);guardatoff=read_eeprom(11);

if (guardaton>0 && guardaton<=20) {incton=guardaton; inctoff=guardatoff;}
```

```

while (1) {

ton=incton; toff=inctoff;

if (!input(pin_b1)) {flag1=1;}

if (incton<20 && flag1==1 && input(pin_b1) ) {flag1=0;++incton;--inctoff;output_high(led);} //se não
//chegou no máximo (incton<50),

write_eeprom(10,incton);write_eeprom(11,inctoff); //se o botão foi pressionado (flag1==1) e se o
//botão já foi solto (input(pin_b1)) incremente

        } // a onda quadrada e guarde os valores na eeprom

if (!input(pin_b2)) {flag2=1;}

if (inctoff<20 && flag2==1 && input(pin_b2) ) {flag2=0;++inctoff;--incton;output_low(led);

write_eeprom(10,incton);write_eeprom(11,inctoff);

        }

output_high(motor);

while(ton) {--ton;delay_ms(1); } //Parte alta da onda quadrada

output_low(motor);

while(toff) {--toff;delay_ms(1); } //Parte baixa da onda quadrada

}}

```

A figura abaixo mostra o circuito montado para com este exemplo. Veja o funcionamento desse circuito no vídeo <http://video.google.com/videoplay?docid=-842251948528771304&hl=en#> .

**PWM com SanUSB e Mosfet IRF540**

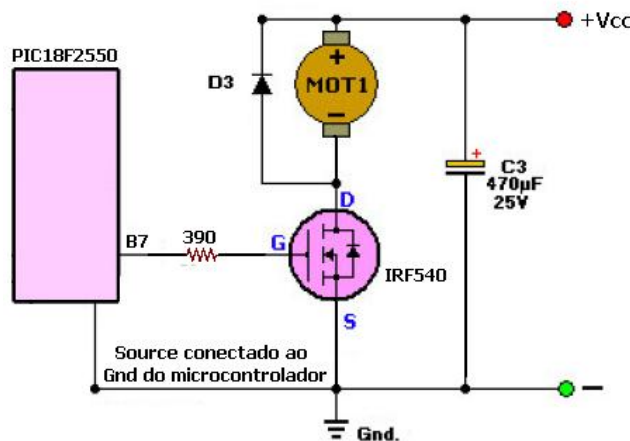


Figura 3.13 – PWM com PIC e IRF540

**Ponte H**

O acionamento da ponte H permite o movimento do motor nos dois sentidos. A ponte H pode ser feita com transistores MOSFETs (mais aconselhável devido à baixa queda de tensão) ou transistores de potência Darlington TIP ou BD.

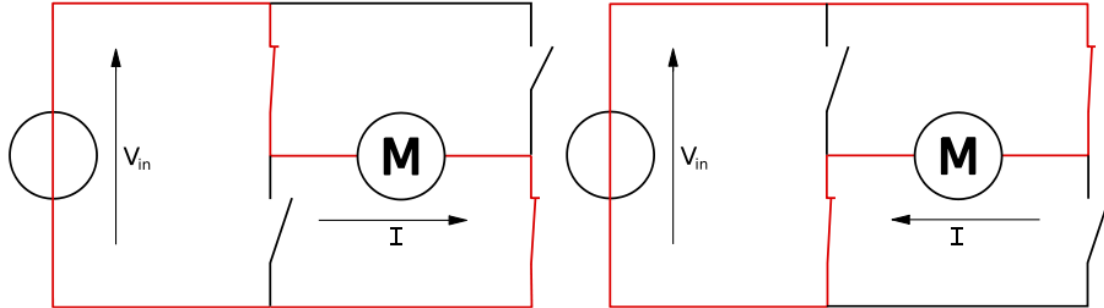


Figura 3.14 – Diagrama de ponte H para motor CC

**PONTE H COM PWM:**

O exemplo abaixo mostra o controle de velocidade de um motor CC por PWM nos dois sentidos com uma ponte H, o incremento da onda quadrada de tensão é feito no pino B2 e o decremento no pino B3. O sentido de rotação é invertido com um botão no pino B5. Na prática no lugar dos transistores é utilizado o *driver* ULN2803 que contém internamente oito transistores com diodos de roda livre. As figuras abaixo mostram a foto do circuito de simulação deste exemplo que pode ser visto em <http://www.youtube.com/watch?v=cFTYHBTEBh8>.

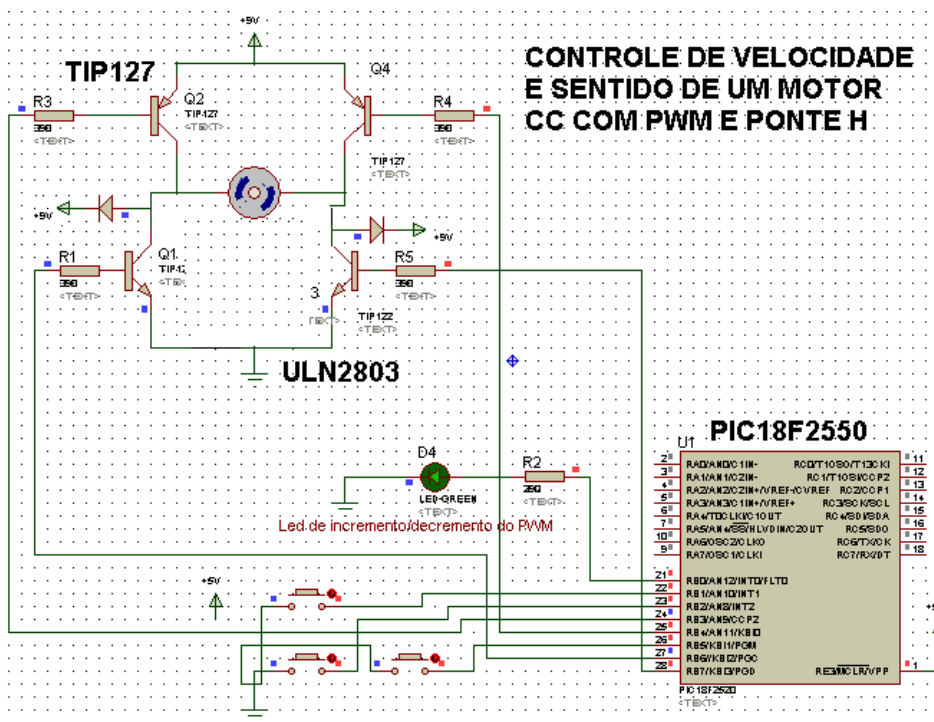


Figura 3.15 – Ponte H com PWM de motor CC



**Programa:**

```
#include <SanUSB.h>

#define esquerdavcc pin_b3
#define direitavcc pin_b4
#define esquerdagnd pin_b6
#define direitagnd pin_b7
#define led pin_b0

unsigned int16 i;
unsigned int ton,toff,incton,inctoff,guardaton,guardatoff;
int1 flag1, flag2, flag3;

main() {
  OSCCON=0B01100110;

  port_b_pullups(true);
  incton=2; inctoff=18; //Período de 20ms - Passo mínimo de tempo = 1ms (5% (1/20) do duty cycle )

  guardaton=read_eeprom(10);guardatoff=read_eeprom(11);
  if (guardaton>0 && guardaton<=20) {incton=guardaton; inctoff=guardatoff;}

  while (1) {ton=incton; toff=inctoff;

  if (!input(pin_b1)) {flag1=1;}
  if (incton<20 && flag1==1 && input(pin_b1) ) {flag1=0;++incton;--inctoff;output_high(led);//se não
  chegou no máximo (incton<50),
  write_eeprom(10,incton);write_eeprom(11,inctoff); //se o botão foi pressionado (flag1==1) e se o
  botão já foi solto (input(pin_b1)) incremente
  } // a onda quadrada e guarde os valores na eeprom

  if (!input(pin_b2)) {flag2=1;}
  if (inctoff<20 && flag2==1 && input(pin_b2) ) {flag2=0;++inctoff;--incton;output_low(led);
  write_eeprom(10,incton);write_eeprom(11,inctoff);
  }

  if (!input(pin_b5)) { // Bateu recuou
    for(i=0; i<400; ++i) //Volta 400 períodos de 20ms
      { ton=incton; toff=inctoff;

      output_high(esquerdavcc);output_low(direitagnd);
      output_low(direitavcc);output_high(esquerdagnd);
      while(ton) {--ton;delay_ms(1); } //Parte alta da onda quadrada

      output_high(direitavcc);output_low(esquerdagnd);
      while(toff) {--toff;delay_ms(1); }
      }

    output_high(direitavcc);output_low(esquerdagnd); //Bloqueia o outro lado
    output_low(esquerdavcc);output_high(direitagnd);
    while(ton) {--ton;delay_ms(1); } //Parte alta da onda quadrada
```

```
output_high(esquerdavcc);output_low(direitagnd);
while(toff) {--toff;delay_ms(1); } //Parte baixa da onda quadrada
}}
```

**Driver ponte H L293D**

Uma das soluções mais simples e barata em atuação de robôs móveis consiste utilizar um integrado *motor driver* como o L293D. Este integrado possibilita o controle de dois motores CC, utilizando quatro pinos de saída do microcontrolador.

O circuito integrado L293D deve ter duas alimentações. Uma para comando (5V) no pino 16 e outra para potência (por exemplo 9,6 V ou 5V) no pino 8. Os motores percebem uma queda de 0,7V em relação à tensão da fonte externa.

As entradas nos pinos 2 e 7 são para acionar o motor A e entradas nos pinos 10 e 15 são para acionar o motor B. O pino 8 é conectado à fonte de alimentação dos motores que tem o mesmo Gnd do circuito de controle.

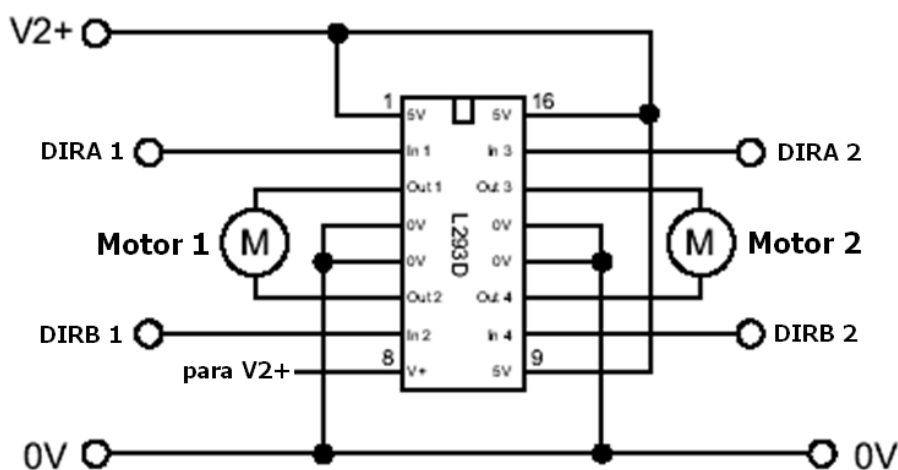


Figura 3.16 – Driver L293D (ponte H)

A mudança dos sinais nos pinos de entrada tem o efeito de produzir a alteração do sentido da corrente no enrolamento do motor, logo do seu sentido de rotação. A Tabela permite programar o movimento em qualquer direção (conjugando 2 motores).

ENABLE	DIRA	DIRB	FUNÇÃO
H	H	L	Para Frente
H	L	H	Para trás
H	L/H	L/H	Stop Rápido
L	X	X	Stop Lento

Figura 3.17 – Comandos de movimento

Se a primeira entrada alta, segunda entrada baixa , então o motor se desloca para frente, se a primeira entrada baixa e a segunda entrada alta , o motor se movimentar para trás. Se ambas as entradas baixas ou altas, o motor pára.

### Solenóides e relés

Uma solenóide consiste num êmbolo de ferro colocado no interior de uma bobina (indutância) elétrica, enrolada em torno de um tubo. Quando se alimenta eletricamente a bobina, cria-se um campo magnético que atrai o êmbolo (núcleo móvel) para dentro da bobina como é mostrado na figura abaixo. No caso de um relé, fecha um contato para circulação de outro nível maior de corrente.

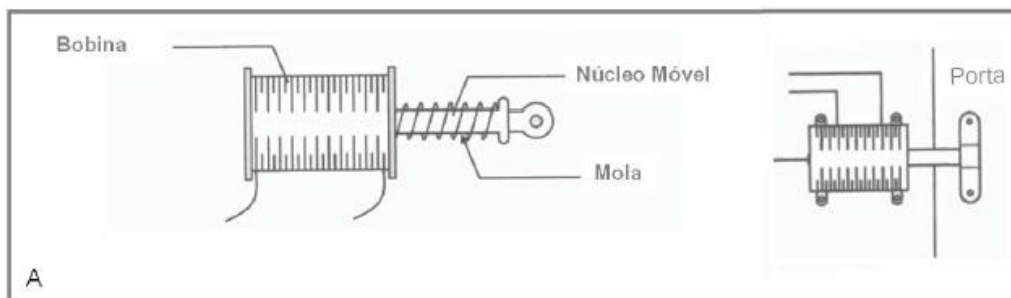


Figura 3.18 – Bobina solenóide

Os relés são dispositivos comutadores eletromagnéticos. A estrutura simplificada de um relé é mostrada na figura abaixo e a partir dela é explicado o seu princípio de funcionamento.

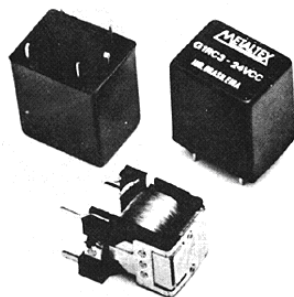


Figura 3.19 – Micro-relés

O controle de uma solenóide ou relé pode ser feito pelo chaveamento de um transistor Darlington ou de um MOSFET mostrado abaixo.

O relé é um tipo de interruptor acionado eletricamente que permite o isolamento elétrico de dois circuitos. O relé é formado por um eletroímã (uma bobina enrolada sobre um núcleo de material ferromagnético) que quando acionado, através da atração eletromagnética, fecha os contatos de um interruptor. Normalmente o interruptor de um relé tem duas posições, com isso existem dois tipos, os NF(normalmente fechado) e NA (normalmente aberto), como mostra a figura abaixo. A bobina do relé é acionada por uma tensão contínua que é especificada de acordo com o fabricante, bobinas de 5, 12 e 24 Volts são as mais comuns.

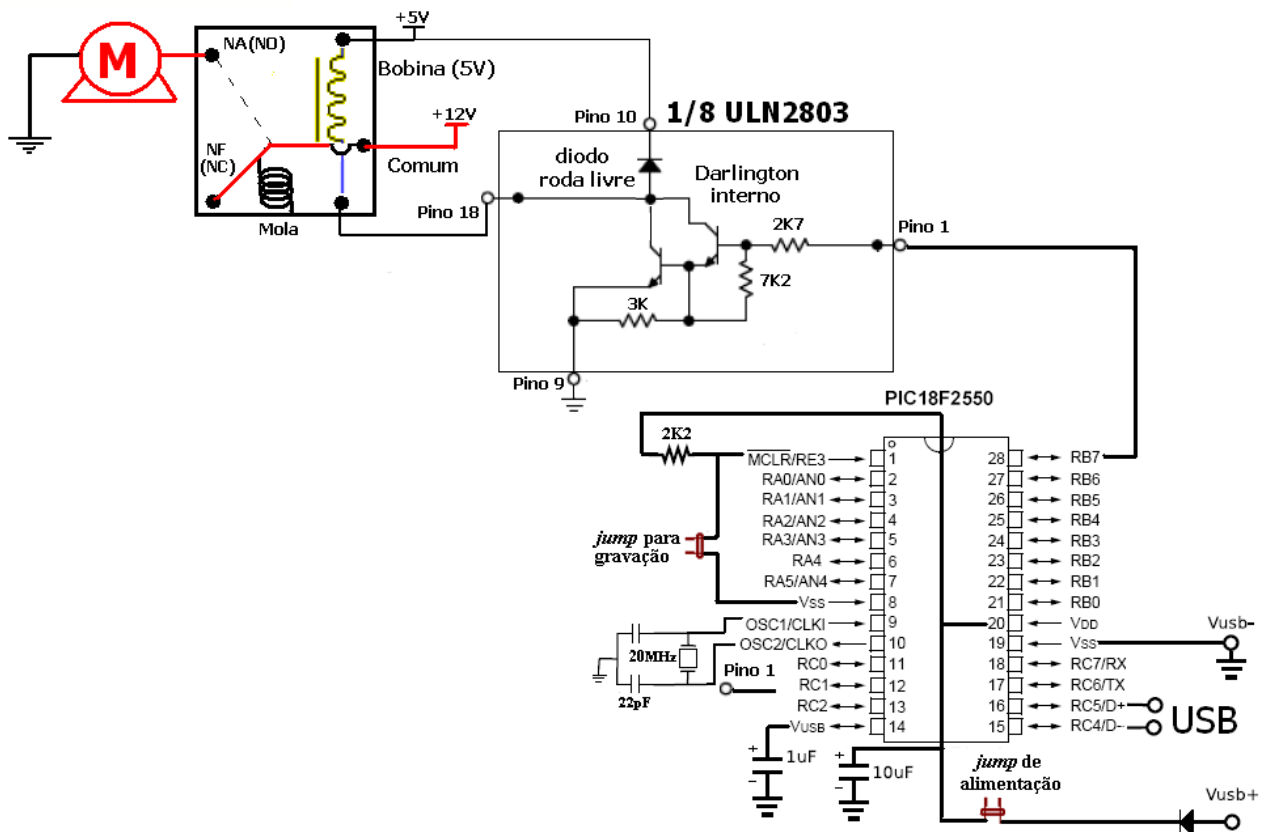


Figura 3.20 – Acionamento de motor com relé 5V e driver ULN2803

Uma das características do relé é que ele pode ser energizado com correntes muito pequenas em relação à corrente que o circuito controlado exige para funcionar. Isso significa a possibilidade de controlar circuitos de altas correntes como motores, lâmpadas e máquinas industriais, diretamente a partir de microcontroladores.

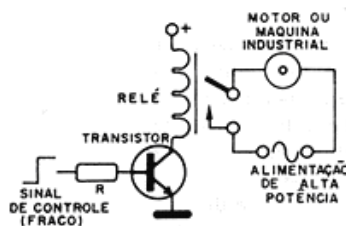


Figura 3.21 – Esquemático interno de um relé e seu acionamento

### Driver de potência ULN2803

Um *driver* de potência é utilizado sempre quando se necessita acionar um *hardware* específico de maior potência. Esse *driver* pode ser usado para controlar motores de passos, solenóides, relés, motores CC e vários outros dispositivos. Ele contém internamente 8 transistores Darlington NPN de potência, oito resistores de base de 2K7 e oito diodos de roda livre, para descarregar no Vcc (pino 10) a corrente reversa da força contra-eletromotriz gerada no chaveamento dos transistores, protegendo os mesmos.

Quando o microcontrolador coloca +5V (nível lógico 1) no pino 1 do *driver* ULN2803, ele conecta o pino 18 do outro lado, onde está ligado um pólo do motor, ao Gnd (nível lógico 0, por isso a simbologia de porta inversora na figura abaixo). Como o outro lado da bobina (o

comum) ou do motor deve estar ligado ao Vcc da fonte de até 30V, esse comando irá energizar a bobina ou o motor.

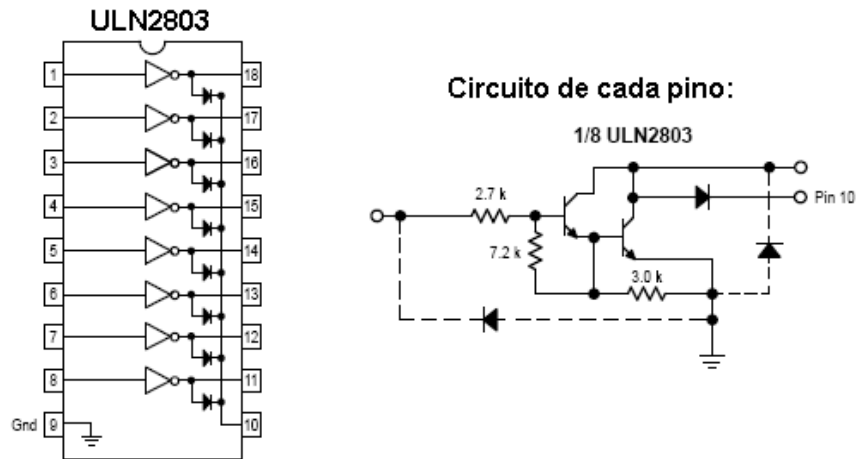


Figura 3.22 – Ligação interna do ULN2803

**Ponte H com micro-relés**

Como foi visto, o acionamento da ponte H permite o movimento do motor nos dois sentidos. A ponte H pode ser feita também com apenas dois micro-relés. Neste caso, pode-se utilizar também o *driver* ULN2803 para a energização das bobinas, pois já contém internamente oito transistores com resistores de base e oito diodos de roda livre. Esse tipo de ponte H, mostrada na figura abaixo, não causa queda de tensão na fonte de alimentação do motor, porque as fontes de energização da bobina do micro-relé e de alimentação do motor devem ser diferentes, ou seja, isoladas uma da outra, para que seja possível o chaveamento do relé.

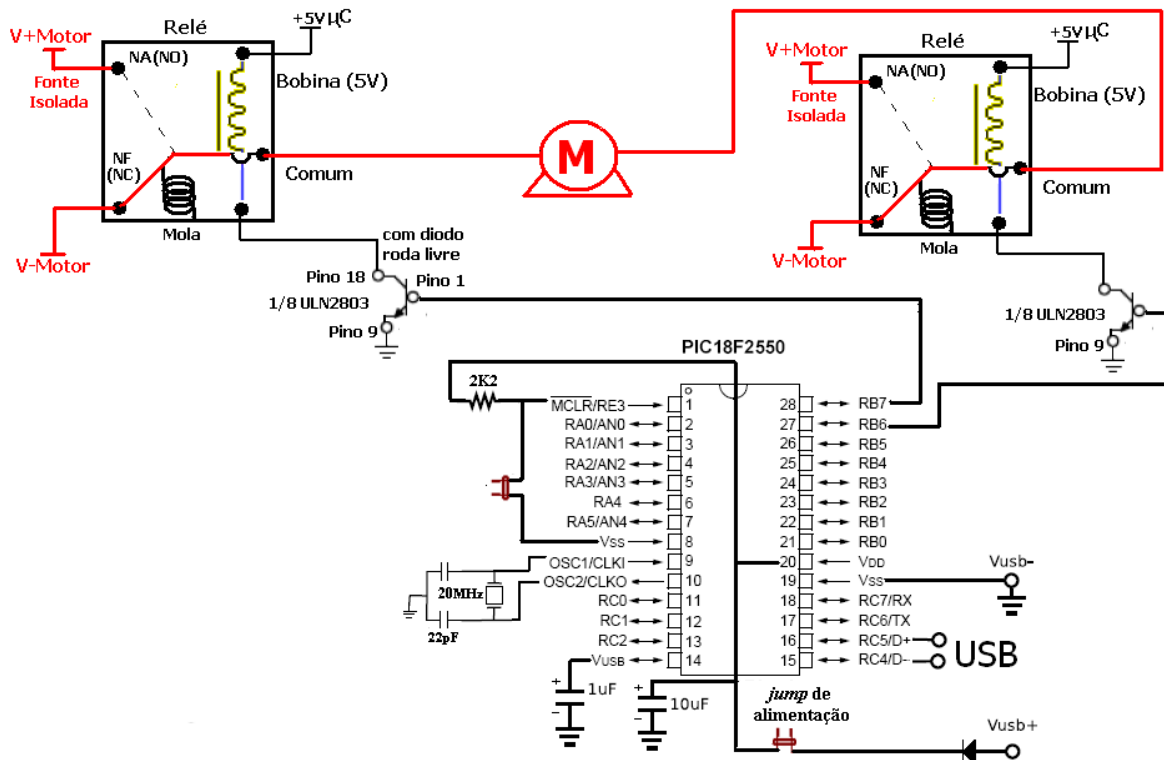


Figura 3.23 – Ponte H de motor CC com micro-relés e ULN2803

Note que inicialmente os dois relés estão conectados ao V-Motor. Ao energizar a bobina do relé da esquerda, conectando o V+Motor, a corrente da fonte passa pelo motor no sentido da esquerda para a direita o que determina o sentido de rotação do motor. Ao desligar o relé da esquerda e acionar o relé da direita ocorre o sentido de rotação inverso do motor.

Quando se utiliza motor CC em ponte H para atuadores robóticos, como rodas de veículos ou braços mecânicos, o que determina o torque e a velocidade do atuador é a relação de transmissão da caixa de engrenagens conectada ao motor.

### 3.2 ACIONAMENTO DE MOTORES DE PASSO

Motores de passos são dispositivos mecânicos eletromagnéticos que podem ser controlados digitalmente.

A crescente popularidade dos motores de passo se deve à total adaptação desses dispositivos à lógica digital. São encontrados não só em aparelhos onde a precisão é um fator muito importante como impressoras, plotters, scanners, *drivers* de disquetes, discos rígidos, mas também, como interface entre CPUs e movimento mecânico, constituindo, em suma, a chave para a Robótica.

#### Motores de passo unipolares

Os motores de passo unipolares são facilmente reconhecidos pela derivação ao centro das bobinas. O motor de passo tem 4 fases porque o número de fases é duas vezes o número de bobinas, uma vez que cada bobina se encontra dividida em duas pela derivação ao centro das bobinas (comum).

Normalmente, a derivação central das bobinas está ligada ao terminal positivo da fonte de alimentação (Vcc) e os terminais de cada bobina são ligados alternadamente à terra através de chaveamento eletrônico produzindo movimento.

As bobinas se localizam no estator e o rotor é um ímã permanente com 6 pólos ao longo da circunferência. Para que haja uma maior resolução angular, o rotor deverá conter mais pólos.

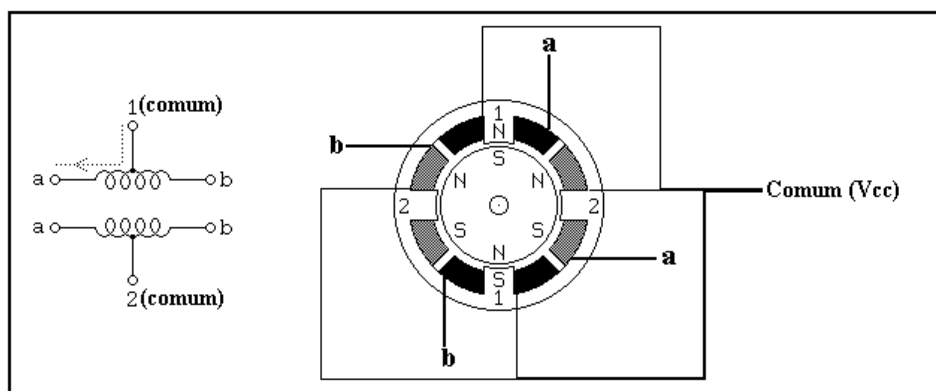
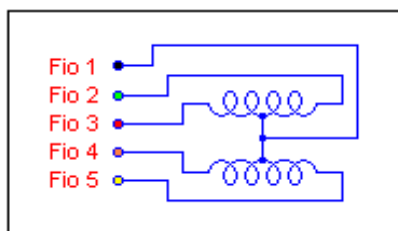


Figura 3.24 – Esquemático interno de um Motor de Passo

Os motores de passo unipolares mais encontrados possuem 5 ou 6 fios. Os motores de passo unipolares de 6 fios possuem **dois fios comuns (derivação central)**. Para o acionamento

do motor de passo, estes fios comuns devem ser ligados à fonte de alimentação (+5V ou +12V) e os terminais da bobina ligados ao controle de chaveamento do motor de passo.

**Motor de 5 fios**



**Motor de 6 fios**

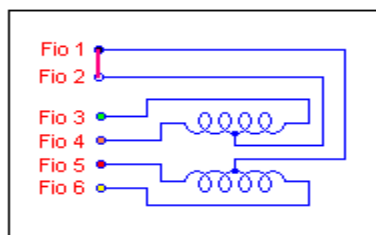


Figura 3.25 – Ligação dos fios dos motores de passo

Para descobrir os terminais de um motor de passo, deve-se considerar que:

Para motores de 6 fios, a resistência entre os fios comuns (Fio 1 e Fio 2) é infinita por se tratarem de bobinas diferentes.

- A resistência entre o fio comum (Fio 1) e o terminal de uma bobina é a metade da resistência entre dois terminais desta bobina.
- Para encontrar a seqüência correta dos fios para chaveamento das bobinas, pode-se ligar manualmente o fio comum ao Vcc, e de forma alternada e seqüencial, o GND (terra) da fonte aos terminais das bobinas, verificando o movimento do motor de passo.

### MODOS DE OPERAÇÃO DE UM MOTOR DE PASSO UNIPOLAR



#### PASSO COMPLETO 1 (FULL-STEP)

-Somente meia bobina é energizada a cada passo a partir do comum;

-Menor torque;

-Pouco consumo de energia.

Nº do passo	1a	2a	1b	2b	Decimal
1-->	1	0	0	0	8
2-->	0	1	0	0	4
3-->	0	0	1	0	2
4-->	0	0	0	1	1





**PASSO COMPLETO 2 (FULL-STEP 2)**

- Duas meia-bobinas são energizadas a cada passo;
- Maior torque;
- Consome mais energia que o Passo completo 1.

Nº do passo	1a	2a	1b	2b	Decimal
1-->	1	1	0	0	8
2-->	0	1	1	0	4
3-->	0	0	1	1	2
4-->	1	0	0	1	1

**Acionamento bidirecional de dois motores de passo**

Como o *driver* de potência ULN2803 ou ULN2804 possui internamente 8 transistores de potência ele é capaz de manipular dois motores de passo ao mesmo tempo. Ele contém internamente oito diodos de roda livre e oito resistores de base dos transistores, o que possibilita a ligação direta ao microcontrolador e aos motores de passo.

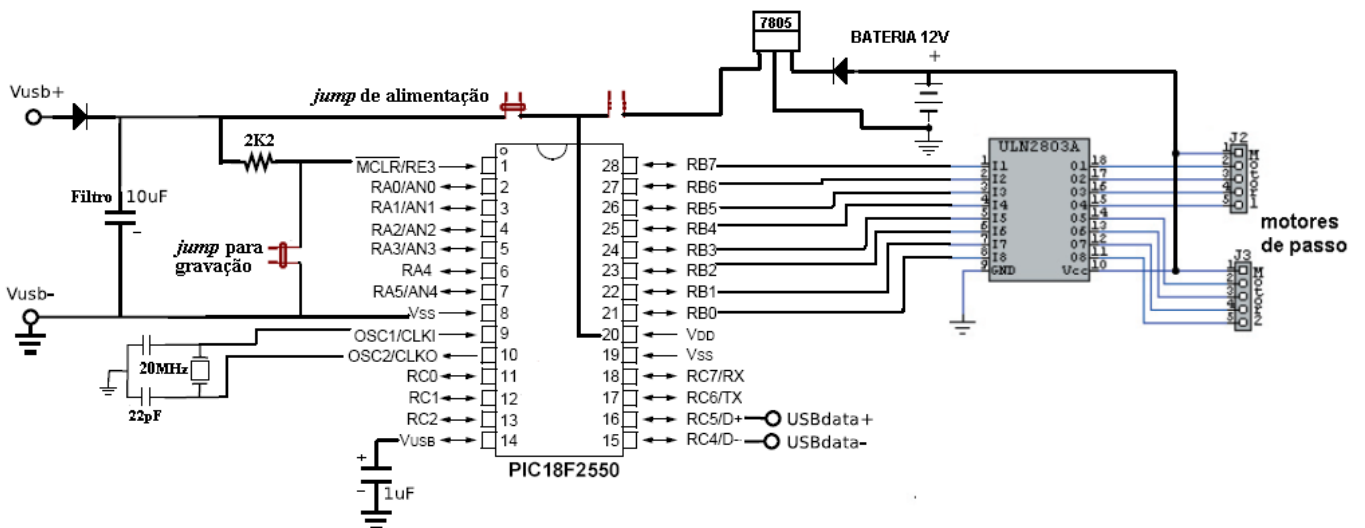


Figura 3.26 – Montagem de circuito com PIC e ULN2803 para acionamento de motor de passo

A bateria para motores de passo deve ter uma corrente suficiente para energizar as bobinas do motor de passo. Dessa forma, é possível associar baterias 9V em paralelo para aumentar a corrente de energização ou utilizar baterias de *No-Breaks*. O link seguinte mostra esse motor utilizando a ferramenta SanUSB: <http://www.youtube.com/watch?v=vaegfA65Hn8>.

### 3.3 SERVO-MOTORES

Há dois tipos de servos: os de posição, com giro de 180º, e o de rotação, que possui o giro contínuo. O Servo de Posição é utilizado em antenas parabólicas, em braços robóticos, na robótica móvel terrestre com o controle de pernas mecânicas e no controle de câmeras. O Servo de Rotação é prioritariamente escolhido para a locomoção por rodas.



Figura 3.27 – Exemplo de servo-motor utilizado em robótica

Trata-se de dispositivos muito precisos que giram sempre o mesmo ângulo para um dado sinal. Um servo típico possui três fios de ligação, normalmente preto, vermelho e branco (ou amarelo). O condutor preto é a referência de massa da alimentação (0 volts), o condutor vermelho é a alimentação e o condutor branco (ou amarelo) é o sinal de posicionamento, como é mostrado na figura abaixo que é um carro para desvio de obstáculos, acionado por dois servo-motores de rotação. O sinal do servo-motor de posição é normalmente um pulso de 1 a 2 milisegundos (ms), repetido depois de um pulso de 10 a 20ms. Com o pulso de aproximadamente 1 ms o servo move-se para um sentido e com o impulso de 2 ms para o sentido oposto. Desse modo, com um impulso de 1,5 ms, o servo roda para a posição central.

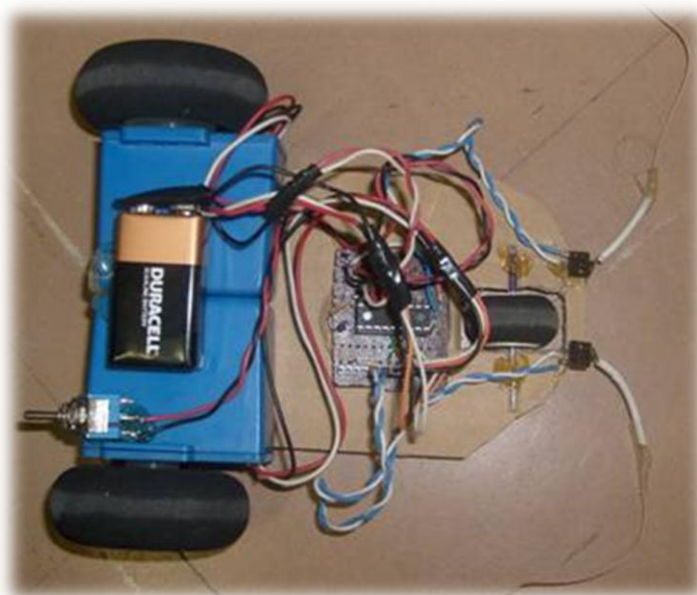


Figura 3.28 – Robô montado com 2 servo-motores de parabólica (abaixo da bateria)

A tensão de alimentação do servomotor é tipicamente de 5V, podendo variar entre 4,5V e 6V. Devido à alta redução do jogo de engrenagens, o torque que se obtém de um servo é bastante alto, considerando o seu tamanho reduzido. Lamentavelmente, os servos consomem correntes elevadas (de 200 mA a 1 A) e introduzem ruído elétrico nos condutores de alimentação, necessitando a aplicação de capacitores de filtro. O programa abaixo move um servo-motor de rotação para frente e um outro para trás. Note que essa operação é utilizada por robôs móveis que possuem dois motores em anti-paralelo.

Programa em linguagem C:

```
#include <SanUSB.h>

#define motor1 pin_b5
#define motor2 pin_b6
int16 frente=50;
short int led;
main(){
clock_int_4MHz();
while (TRUE)
{
while (frente>0)
{ output_high(motor2); //Inicializa o pulso do motor 1
output_high(motor1); //Inicializa o pulso do motor 2
delay_ms(1);
output_low(motor1); //Termina o pulso de 1ms do motor1– sentido horário
delay_ms(1);
output_low(motor1);
output_low(motor2); //Termina o pulso de 2ms do motor2 – sentido anti-horário
delay_ms(10);
--frente;
}
frente=50;
led=!led; //pica led a cada 50 ciclos do servo-motor, ou seja a cada 12ms*50 = 600ms
output_bit(pin_b7,led);
}}
```

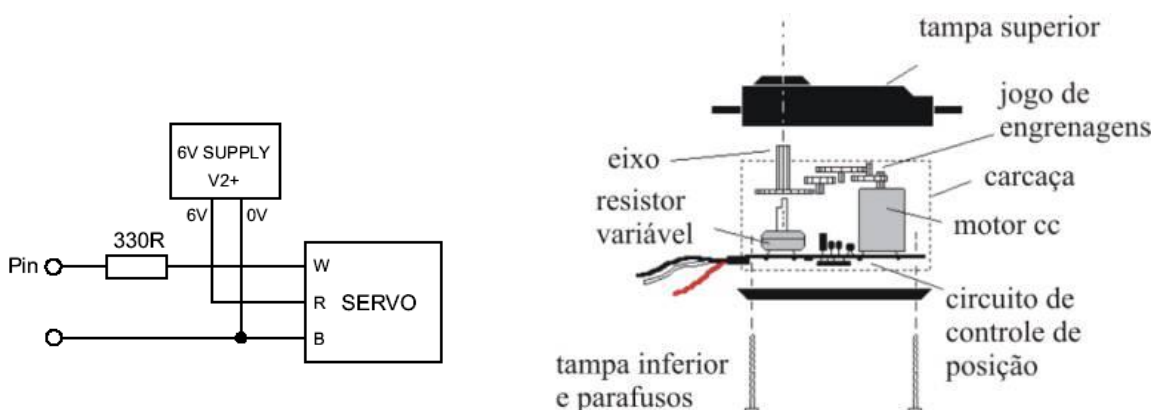


Figura 3.29 – Ligação do servo-motor e partes internas

O programa abaixo move um servo-motor de posição. Com o pulso de aproximadamente 1 ms, o servo move-se para 0 graus, e com o pulso de 1,2 ms para 90 graus e com o pulso de 2 ms para 180 graus. Note que este motor é utilizado em antenas parabólicas, em sistemas robóticos e no controle de câmeras.

```
#include <SanUSB.h> //Servo_motor de parabólica - Apresenta 3 posições definidas 0, 90 e 180 graus

int16 FRENTE=200, TRAS=200; //no servo de teste 200 passos corresponderam a aprox. 1 volta

void main()
{
  clock_int_4MHz();

  while (1)
  {

    while (FRENTE>0)
    {
      FRENTE--;
      output_high(pin_b0);
      delay_ms(1); // tempo de busca de 0 graus
      output_low(pin_b0);
      delay_ms(10);
    }
    FRENTE=200;

    while (TRAS>0)
    {
      TRAS--;
      output_high(pin_b0);
      delay_ms(2); // tempo de busca de 180 graus
      output_low(pin_b0);
      delay_ms(10);
    }
    TRAS=200;

    output_toggle(pin_b7);

  }
}
```

---

#### 4. SENSORES

Os sensores auxiliam o operador no direcionamento do robô ao longo da trajetória através de transmissão de informações ao controlador. Com isso torna-se mais caro que os outros métodos, como o manual, uma vez que utiliza equipamentos mais sofisticados. Contudo, os sensores são utilizados somente durante o processo de aprendizagem, uma vez que na execução da tarefa não serão afetados por interferência do robô.

Com a utilização dos sensores há uma diminuição do tempo e da força necessária para ensinar o robô a realizar as tarefas.

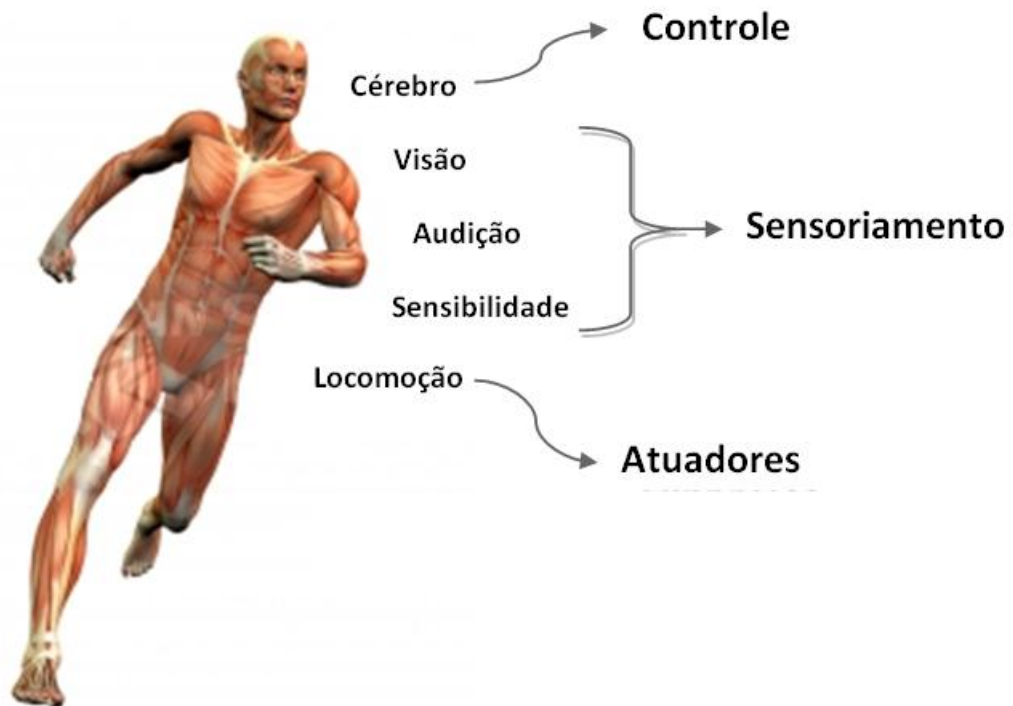
Os sensores têm a finalidade de permitir a monitoração do robô, ou seja, são dispositivos projetados para perceber ou detectar parâmetros especificados por meio do controlador, a partir de sua relação com o mundo e com a tarefa que lhe foi determinada. São estes tipos de sensores que funcionam como os órgãos da “visão”, “audição” e “tato” dos robôs, e permitem que estes se orientem no mundo.

Os principais problemas enfrentados no desenvolvimento da robótica móvel surgem diretamente da necessidade de interação com objetos físicos e entidades do ambiente. Esta interação consiste principalmente na localização e identificação destas entidades e objetos enquanto o robô move-se (navega) neste meio ambiente. Este processo precisa ser realizado rapidamente, e com a máxima precisão e acurácia.

No projeto de tais sensores, deve-se levar em conta as seguintes considerações :

- **Campo de visão:** deve ser amplo o bastante e com suficiente profundidade para adequar-se à sua aplicação;
- **Range Capability (Capacidade de Alcance):** o alcance mínimo de detecção, bem como o alcance máximo efetivo, devem ser adequados ao uso pretendido do sensor;
- **Acurácia e Resolução:** devem estar de acordo com as necessidades da tarefa dada;
- **Habilidade para detectar todos os objetos no ambiente:** várias condições ambientes podem interferir e acrescentar ruído às informações captadas pelos sensores. Os mesmos devem ser capazes de minimizar tais interferências;
- **Operação em tempo-real:** o sensor deve ser capaz de fornecer dados em tempo real a velocidades que sejam compatíveis com a velocidade de sua plataforma receptora (robô móvel, veículo-robô, etc.);
- **Dados concisos e de fácil interpretação:** o formato de saída de um sensor deve ser realístico do ponto de vista dos requisitos de processamento, visando reduzir ao máximo o pré-processamento da informação fornecida pelo sensor.
- **Redundância:** o sistema sensor deve sofrer uma degradação lenta e suave; e não ficar totalmente incapacitado devido à perda de um elemento sensor;
- **Simplicidade:** o sistema deve ser de baixo custo e modular, para facilitar a manutenção e permitir atualizações revolucionárias, portanto, não deve estar restrito a um tipo de hardware específico;
- **Consumo de força:** o consumo de força deve ser mínimo para poder ser mantida uma boa relação com os recursos limitados a bordo de robôs móveis e veículos-robôs.
- **Tamanho e peso:** devem ser o menor possível, de acordo com os objetivos pretendidos para o robô.

Estas considerações não apresentam regras rígidas, mas o projetista deve sempre tê-las em mente antes de iniciar o projeto de um novo sistema sensor para um robô móvel.



#### 4.1 SENSORES DE PROXIMIDADE

Servem para indicar ao robô se ele está próximo de algum objeto do mundo. Este conceito de proximidade geralmente está relacionado a algum intervalo de distância especificado. São divididos em sensores de contato e de não-contato.

##### Sensores de Contato

Sensores tácteis (ou de contato) são geralmente empregados em veículos guiados automaticamente (*Automated Guided Vehicle - AVGs*) e em robôs móveis para fornecer uma indicação de colisão de último recurso com obstruções em torno do robô. O processo de detecção deste tipo de sensor envolve contato físico direto entre o sensor e o objeto de interesse.

Atualmente várias tecnologias têm sido aplicadas na construção dos chamados esquemas sensíveis tácteis. Entre essas tecnologias de sensores, destacamos os seguintes como mais representativos:

- Chaves de contato
- Magnéticos
- Piezoelétricos
- Capacitivos
- Fotoelétricos
- Magneto-resistivos
- Piezo-resistivos
- Ultrasônicos

Do ponto de vista da robótica móvel, estes esquemas sensores podem ser divididos em três grupos principais:

1. Sensores tácteis ou antenas
2. Amortecedores tácteis
3. Arrays de superfície distribuídos

**Sensores Tácteis ou Antenas:** consistem, geralmente, de hastes de metal ligadas a terminais de chaves de contato. A figuras 4.1 e 4.2 mostram tipos simples de sensores tácteis. Estes podem ser divididos em duas categorias:

- **Sensores passivos:** que são geralmente fixos e
- **Sensores ativos:** que geralmente implementam alguma estratégia mecânica de busca para tornar a orientação mais efetiva. Este processo assemelha-se muito ao movimento das antenas de um inseto.

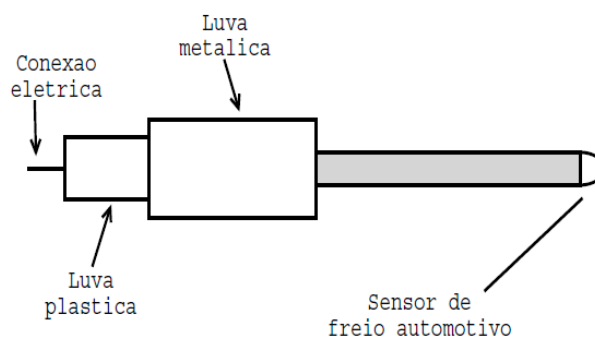


Figura 4.0 – Exemplo de sensor táctil passivo

Atuador

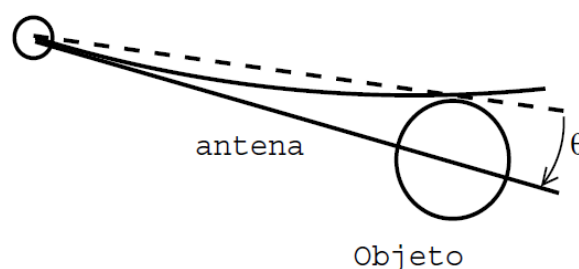


Figura 4.1 – Sensor táctil ativo em forma de antena

**Amortecedores Tácteis:** consistem, geralmente, de uma placa de metal presa por uma ponta e curvada para fora na outra, formando um amortecedor protetor na frente do veículo. Esta placa está restrita a um único grau de liberdade de movimento, e quando ela sofre pressão externa provoca a ativação de alguma chave de contato.



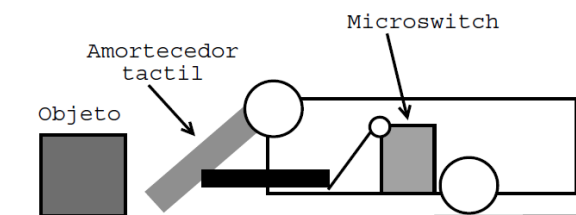


Figura 3.2 – Exemplo de amortecedor tátil

**Arrays de superfície distribuídos:** são a nova tendência na construção de sensores tácteis. São arranjos complexos de sensores, com o objetivo de fornecer o perfil do objeto em contato com o array sensor. Além do perfil bidimensional do objeto, este array também é capaz de fornecer alguma quantificação da magnitude da força do contato. Atualmente tem surgido um grande interesse pelo desenvolvimento de sensores contínuos, como “peles”, os quais poderiam ser incorporados diretamente ao exterior dos robôs ou veículos robóticos.

### Sensores de Proximidade de não contato

São utilizados para determinar a presença de objetos próximos. Foram desenvolvidos para estender a faixa sensitiva além daquela permitida pelos sensores de contato. A sua principal vantagem é evitar o contato direto entre o robô e os objetos circundantes. Costumam ser classificados de acordo com as suas propriedades específicas que são usadas para disparar a ação, nas seguintes categorias:

- Magnéticos
- Indutivos
- Ultrasônicos
- Microondas
- Óticos
- Capacitivos

**Sensores de Proximidade Magnéticos:** esta categoria inclui os *reed switches*, dispositivos de efeito *Hall* e sensores magneto-resistivos. *Reed Switches* magnéticos são o tipo mais simples de sensores de proximidade magnéticos. Consistem de um par de *reeds* ferromagnéticos de baixa relutância, fixados em extremidades opostas de um tubo de vidro hermeticamente fechado, dispostos de tal forma que suas pontas se sobrepõem levemente, porém sem se tocarem. As pontas dos filamentos assumem polaridades opostas quando expostas a um fluxo magnético externo, e a subsequente força atrativa força os dois *reeds* a fazerem contato, fechando um circuito elétrico que dispara a ação.



Figura 4.3 – Reed Switch

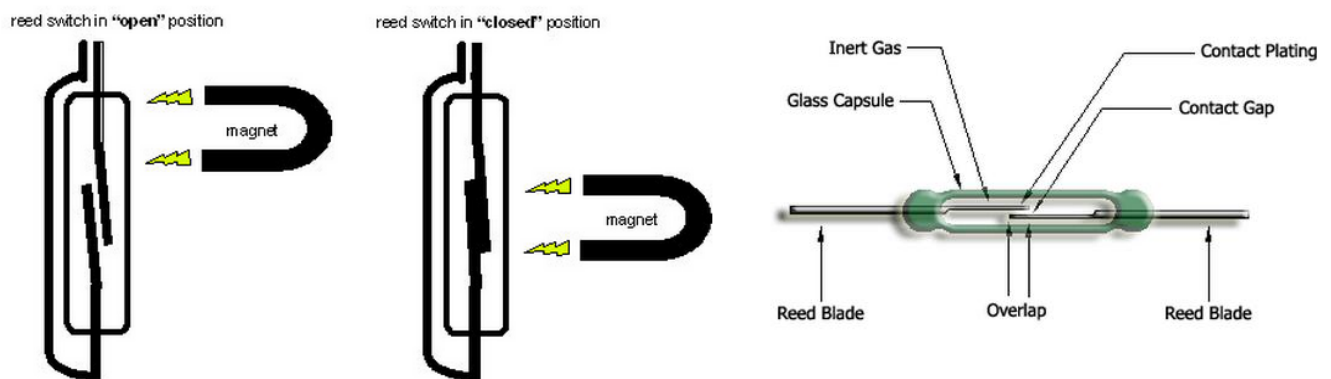


Figura 4.4 – Funcionamento de *reed switch* com aproximação de ímã e descrição das partes

Alguns problemas com este tipo de sensor podem ocorrer devido à trepidações, vibrações estruturais, etc. Estes sensores estão disponíveis em duas versões: normalmente aberto e normalmente fechado, e são comumente empregados como sensores de fechamento de portas e janelas em aplicações de segurança.

**Sensores de Proximidade Ultra-Sônicos:** são exemplos de sensores reflexivos, ou seja, são sensores que respondem a mudanças na quantidade de energia emitida que retorna para um detector após interagir com o objeto alvo. Sistemas ultra-sônicos típicos consistem de dois transdutores (um para transmitir e outro para receber a energia retornada), embora a velocidade relativamente baixa do som permita operar no modo transceiver, com apenas um transdutor.

O transmissor emite uma onda longitudinal na região ultra-sônica do espectro acústico (tipicamente 20 a 200 KHz) acima do limite normal do ouvido humano. A resposta recebida pelo receiver é uma função da amplitude da energia retornada, em oposição ao tempo passado antes da detecção do eco do som emitido.

Sensores ultra-sônicos são bastante úteis a distâncias maiores do que os sensores citados anteriormente, e podem detectar tanto objetos sólidos quanto líquidos. Seu funcionamento é semelhante ao de um radar: se um objeto entra no campo do sinal acústico, a energia é refletida de volta ao receiver. Como acontece com qualquer sensor reflexivo, o alcance máximo de detecção é dependente não apenas do nível de potência do sinal emitido, mas também de certas características do alvo, como área da seção reta, reflectividade e directividade. Uma vez que a amplitude do sinal recebido alcança um limiar pré-estabelecido, o sensor modifica o seu estado de saída para indicar a detecção.

**Sensores de proximidade óticos:** opostos, retroreflectivos e difusos.

**Sensores Óticos Opostos:** conhecidos como “olhos elétricos”, surgiram nos anos 1950, e foram muito utilizados em uma grande variedade de aplicações, como abertura automática de portas, sistemas de segurança, etc. Consiste basicamente de duas unidades separadas - um emissor e um receptor - colocadas na região de interesse (emissor) e em alguma parte do robô móvel (receptor).

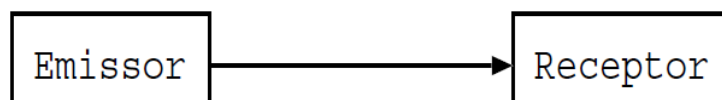


Figura 4.5 – Sensor Óptico Oposto

O emissor emite um fecho de luz (geralmente através de umLED) que é captado pelo receptor (no robô) indicando assim a presença do referido objeto (que possui o emissor). Geralmente tem pouca aplicabilidade em robótica móvel devido à sua configuração geométrica, ou seja, a necessidade de pares opostos e separados de emissor e receptor.

**Sensores Óticos Retro-reflexivos:** podem ser considerados uma evolução dos sensores óticos opostos, pois também possui um par transmissor/receptor, porém estes não estão mais colocados em partes opostas,mas sim apenas em uma única parte (no caso da robótica móvel, o robô).

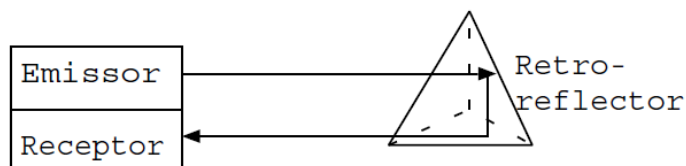


Figura 4.6 – Sensor ótico retro-refletor

O nome Retro-refletivo deve-se ao uso de um retro-refletor (como um espelho) colocado de maneira oposta a um par transmissor/receptor. A percepção de um objeto dá-se então pela reflexão de um raio de luz emitido pelo emissor (instalado no robô) e refletido pelo retro-refletor (fixado no objeto de interesse). São usados em robótica móvel no auxílio à navegação, sendo os retro-refletores utilizados como marcações de posição que são identificadas pelo robô enquanto este se desloca no ambiente.

**Sensores óticos difusos:** operam de forma similar aos retro-reflexivos, com a diferença de que no caso dos sensores difusos não há a presença de uma contra-parte retro-reflexiva, assim, a energia retornada pelo objeto é a difusão da luz na própria superfície do objeto. Estudaremos e aprenderemos a montar mais adiante este tipo de sensor aplicado na prática em um robô móvel.

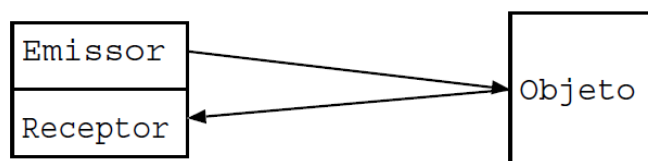


Figura 4.7 – Sensor ótico difuso

## 4.2 TRANSMISSOR E RECEPTOR INFRAVERMELHO (IR)

Os transmissores e receptores IR (*infra-red* ou Infravermelhos) são muito utilizados como sensor ótico por reflexão difusa para registro de posição. A figura abaixo mostra o circuito do sensor e um gráfico do sinal de saída (em mV) do receptor IR em função da distância perceptível pelo receptor IR com resistor de elevação de tensão para 5V (*pull-up* 2K2). O vídeo XXX mostra essa variação, com o acionamento de um led pela queda do sinal analógico através da condução do receptor IR.

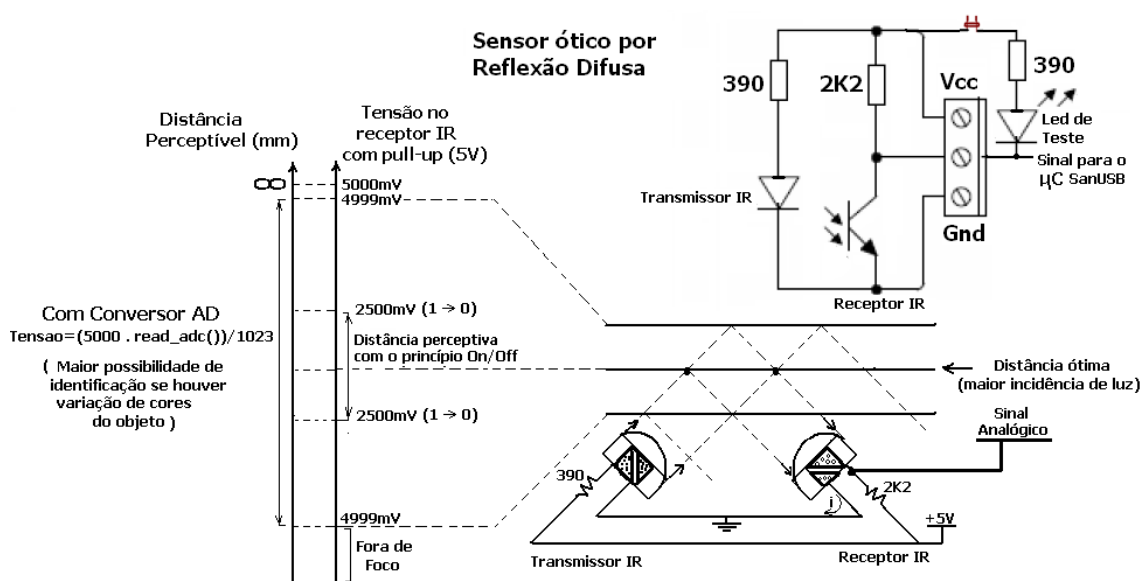


Figura 4.8 – Esquemático de montagem de par de sensor infravermelho

Para ter um maior alcance e perceber uma maior variação de posição com esse sensor, é aconselhável utilizar o conversor AD de 10 bits do microcontrolador para identificar variações de até 5mV no sinal do sensor. A distância ótima é a distância em que incide no receptor a maior parte do feixe de luz emitido pelo transmissor. Nessa distância ocorre a maior variação (queda) do sinal de saída analógico do receptor IR.

Utilizando o princípio *on/off*, só há identificação de posição quando o sinal do receptor for menor ou igual a 2,5V (nível lógico baixo), o que torna o sensoriamento muito limitado.

Durante os testes desse circuito foi verificado que, devido a resistência de 390 Ω em paralelo com 2K2 Ω, quando o led é conectado no circuito, há uma diminuição na variação do sinal de saída analógico do receptor IR. O programa a seguir mostra a leitura em mV do sensor ótico via emulação serial.

```
#include <SanUSB.h> //Leitura de tensão em mV com variação de um potenciômetro
#include <usb_san_cdc.h> // Biblioteca para comunicação serial virtual

int32 tensao;

main() {
clock_int_4MHz();

usb_cdc_init(); // Inicializa o protocolo CDC
usb_init(); // Inicializa o protocolo USB
usb_task(); // Une o periférico com a usb do PC

setup_adc_ports(AN0); //Habilita entrada analógica - A0
setup_adc(ADC_CLOCK_INTERNAL);

while(1){
//ANALÓGICO      DIGITAL(10 bits)
set_adc_channel(0); // 5000 mV      1023
delay_ms(10); // tensao      read_adc()
tensao= (5000*(int32)read_adc())/1023;
printf (usb_cdc_putc,"\r\nA tensao e' = %lu mV\r\n",tensao); // Imprime pela serial virtual

output_high(pin_b7);
delay_ms(500);
output_low(pin_b7);
delay_ms(500);    }}

```

---

## 5. MONTAGEM DO ROBÔ MÓVEL DE BAIXO CUSTO

### Passo a passo da montagem do robô móvel microcontrolado seguidor de labirinto:

1. Uma base semelhante à mostrada na figura 5.0, de acrílico, é utilizada como chassi do veículo-robô. Sobre ela são colocados o circuito de controle (ferramenta SanUSB) e o conjunto de 4 pilhas de 1,5V; nas laterais são fixados com parafusos os servo-motores (em furos feitos nas colunas verticais) e no eixo destes, as rodas. Os fios dos servo-motores passam pelos recortes no topo retangular até o circuito para completar a ligação do sistema robótico. A base pode ter qualquer outra forma e pode ser feita de outros tipos de materiais como: madeira ou alumínio; de forma que se obtenha uma estrutura fixa capaz de sustentar o peso das pilhas e tenha o equilíbrio necessário para executar as ações de forma eficiente.

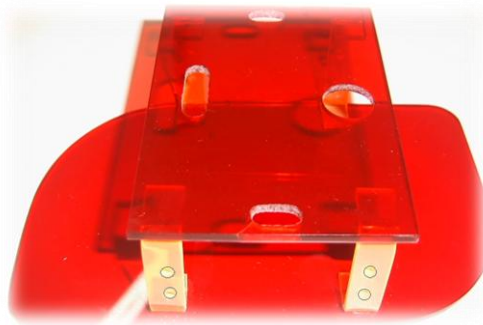


Figura 5.0 – Chassi do robô móvel

2. O primeiro passo é fixar os motores nas colunas verticais. Alinhe os furos dos servos aos furos das colunas como mostra a figura 5.1 e utilizando parafusos semelhantes aos da figura 5.2, fixe os 4 pontos. Lembrando que o eixo do motor deve estar do mesmo lado do furo maior e circular do topo retangular do chassi, como mostra a figura 5.4.

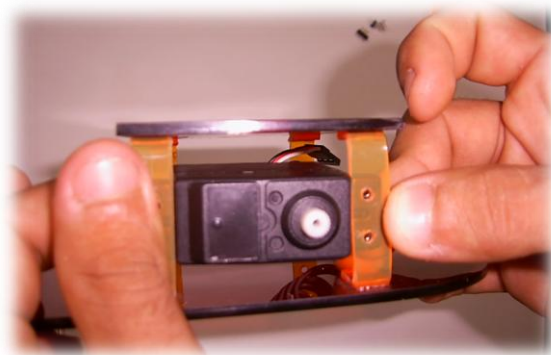


Figura 5.1 – Fixação dos servo-motores



Figura 5.2 – Parafusos para fixação dos servos

Com uma chave apropriada, prossiga com a fixação dos parafusos. Apoie por trás com o dedo (figura 5.3), de forma oposta ao parafuso, para facilitar o rosqueamento deste.

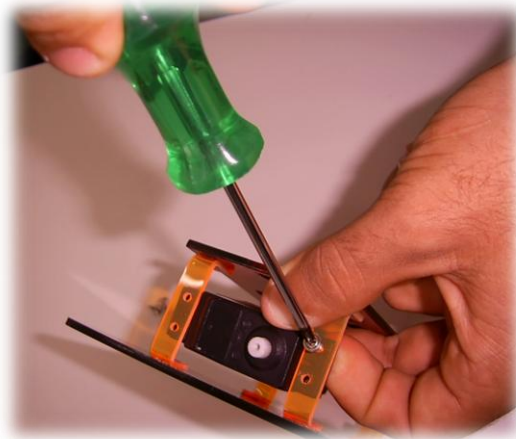


Figura 5.3 – Fixação dos parafusos



Figura 5.4 – Servo-motor esquerdo fixo

Faça o mesmo para o motor direito, de forma que a montagem completa fique como mostra a figura 5.5.



Figura 5.5 – Fixação dos 2 servo-motores



3. Prosseguimos com a fixação da placa de controle. Deve-se rosquear 2 parafusos semelhantes aos da figura 5.6 na base retangular do chassi e posicionar a placa entre estes de forma que fique presa. Coloca-se um pedaço de E.V.A (espécie de borracha em folha) entre a placa e o chassi, de forma a prender com mais segurança e evitar que os pinos dos componentes da placa arranhem o acrílico.

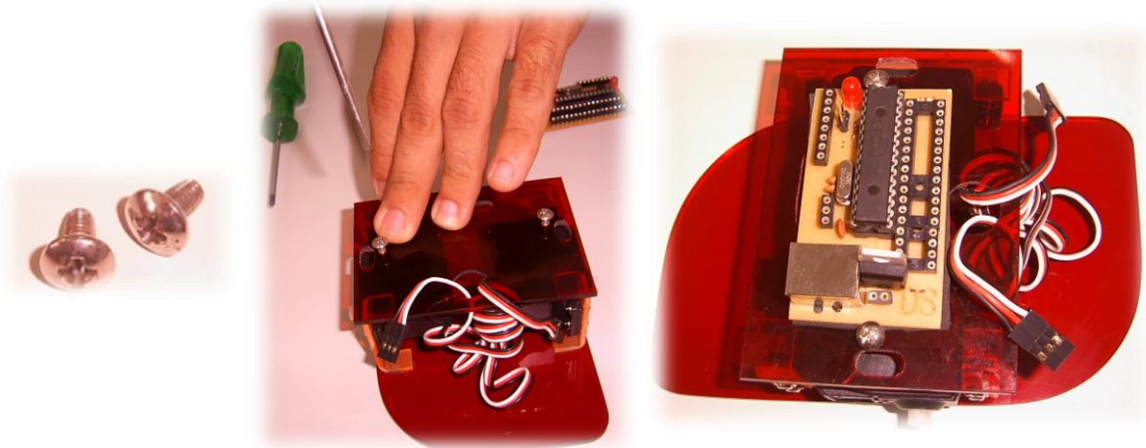


Figura 5.6 – Parafusos e fixação da placa de controle

4. Deve-se agora conectar os fios dos servo-motores (figura 5.7) seguindo orientação existente na placa de controle.

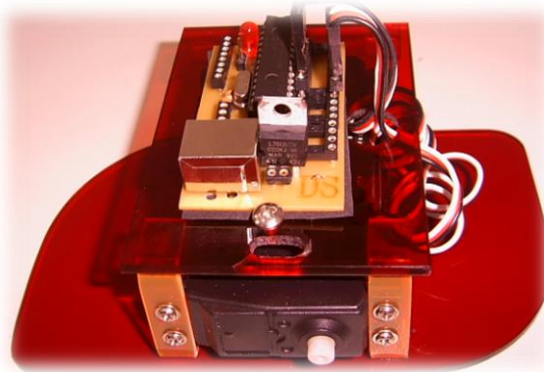


Figura 5.7 – Conexão dos fios dos servos

5. O passo seguinte é a fixação da esfera traseira de apoio. É necessário apenas rosqueá-la no furo da base maior (figura 5.8).



Figura 5.8 Fixação da esfera de apoio

6. Rosqueie em seguida as rodas no eixo dos servo-motores (figura 5.9).



Figura 5.9 – rodas

7. O robô móvel com as rodas fixadas pode ser observado na figura 5.10.

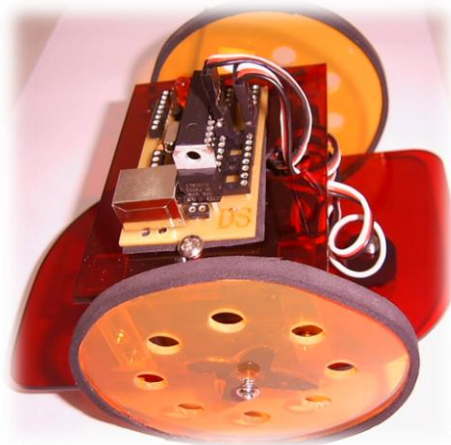


Figura 5.10 – Robô móvel com as rodas fixadas

8. Em seguida deve-se fixar a plaquinha dos sensores infravermelhos (figura 5.11) na base inferior do chassi. Um par deve ser orientado para a parte frontal e outro para a parte lateral. Pode-se utilizar uma proteção para os mesmos como mostra a figura 5.12. Esta protege os sensores contra sujeira e contra interferência de raios infravermelhos advindos de fontes de iluminação artificial (lâmpadas fluorescentes) e natural (sol: fonte de todas as frequências luminosas).



Figura 5.11 – Plaquinha do par de sensores infravermelhos



Figura 5.12 – Proteção dos sensores e fixação destes na base inferior

9. O último passo é o posicionamento das pilhas/baterias (alimentação da placa de controle). Cole com fita ou silicone o estojo de pilhas e conecte os fios à placa como mostra a figura 5.13.



Figura 5.13 – Fixação das pilhas

---

## CONSIDERAÇÕES FINAIS

Utilizando os princípios desta apostila, estudantes foram três vezes consecutivas campeões da Competição de Robótica do IFCE (2007, 2008 e 2009) na categoria Localização, campeões da Feira Brasileira de Ciências e Engenharia (FEBRACE09) da USP em São Paulo na Categoria Engenharia (2009), vice-campeões da Competição de Veículos Mecatrônicos (CVM 2009) na categoria Master, campeões na IX EXCETEC, como também Prêmio de Inovação em Aplicação Tecnológica na Ferie Explora 2009 em Medelin na Colômbia, campeões no Foro Internacional de Ciencia e Ingeniería 2010 no Chile na Categoria Supranivel e campeões de Robótica na categoria Labirinto da FEMECI 2010 (IFCE Campus Cedro).

---

## REFERÊNCIAS BIBLIOGRÁFICAS

- Grupo SanUSB (2011). Arquivos do Grupo SanUSB. Retirado em 05/01/11, no World Wide Web: [www.tinyurl.com/SanUSB/](http://www.tinyurl.com/SanUSB/).
- Jucá, S.C.S. (2011). Apostilila de microcontroladores PIC e periféricos. Retirado em 09/01/11, no World Wide Web: [http://www.4shared.com/document/Qst\\_pem-100923Apostila\\_CPIC.html](http://www.4shared.com/document/Qst_pem-100923Apostila_CPIC.html).
- Jornal O Povo (2011). Da escola pública para o mundo. Retirado em 05/01/11, no World Wide Web: <http://www.opovo.com.br/app/opovo/cienciaesaude/2011/01/08/noticiacienciaesaudejornal,2086691/da-escola-publica-para-o-mundo.shtml>.
- Jornal O Povo (2010). De Maracanaú para Eslováquia. Retirado em 05/01/11, no World Wide Web: <http://publica.hom.opovo.com.br/page,489,109.html?i=2051467>.
- Diário do Nordeste (2010). Robô cearense. Retirado em 05/01/11, no World Wide Web: <http://diariodonordeste.globo.com/materia.asp?codigo=861891>.
- TV Diário (2010). Feira do Empreendedorismo SEBRAE. Retirado em 05/01/11, no World Wide Web: [http://www.youtube.com/watch?v=8Y7gOPd\\_zN4](http://www.youtube.com/watch?v=8Y7gOPd_zN4).
- TV Verdes Mares (2009). Estudantes competem com robôs. Retirado em 05/01/11, no World Wide Web: <http://tvverdesmares.com.br/bomdiaceara/estudantes-competem-com-robos/>.
- TV Cidade (2009). Projetos Comsolid/Setapi IFCE. Retirado em 05/01/11, no World Wide Web: [http://www.youtube.com/watch?v=i\\_waT0\\_201o](http://www.youtube.com/watch?v=i_waT0_201o).